ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

THESIS PRESENTED TO
ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
Ph.D.

BY
Vasile STROIAN

DEVELOPMENT OF A PROTOTYPE FOR MULTIDIMENSIONAL PERFORMANCE MANAGEMENT IN SOFTWARE ENGINEERING

MONTREAL, 19 APRIL 2011

THIS THESIS HAS BEEN EVALUATED

BY THE FOLLOWING BOARD OF EXAMINERS:

Pierre Bourque, Ph.D., Thesis Supervisor
Department of Software and IT Engineering, École de Technologie Supérieure

Alain Abran, Ph.D., Thesis Co-Supervisor
Department of Software and IT Engineering, École de Technologie Supérieure

Alain April, Ph.D., President of the Board of Examiners
Department of Software and IT Engineering, École de Technologie Supérieure

Eric Paquette, Ph.D., Examiner
Department of Software and IT Engineering, École de Technologie Supérieure

Andreas Schmietendorf, Ph.D., External Examiner
University of Magdeburg, Magdeburg, Germany

THIS THESIS WAS PRESENTED AND DEFENDED
BEFORE A BOARD OF EXAMINERS AND PUBLIC

22 MARCH 2011

AT ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

DEDICATION

This thesis is dedicated to my mother Maria and my father Grigore for their endless love and support throughout my life.

This thesis is also dedicated to my daughter, Marie Louise, with the hope that she will realize that higher education is important in life, and that one day her dream to become a medical doctor will come true.

ACKNOWLEDGMENTS

A Ph.D. thesis should bring the writer feelings of fulfillment, accomplishment, and joy, which is certainly the case for me.

I am grateful to the following people, who provided guidance, advice, and support throughout my Ph.D. studies.

I heartily thank my supervisor, Pierre Bourque, Ph.D. for his encouragement, commitment, and support during this lengthy and often stressful process. His professionalism, constructive criticism, and guidance helped me during my research and showed me the way into the world of "engineering science".

I extend my warm and sincere thanks and gratitude to my co-supervisor, Alain Abran, Ph.D. I have been privileged to benefit from his comprehensive expertise and vast knowledge.

I also thank my daughter, Marie Louise, for her support. She has been a shining light throughout these past few years. She always encouraged me, even at the most difficult times.

DEVELOPMENT OF A PROTOTYPE FOR MULTIDIMENSIONAL PERFORMANCE MANAGEMENT IN SOFTWARE ENGINEERING

Vasile STROIAN

RÉSUMÉ

Une meilleure compréhension et l'amélioration de la performance sont des problèmes importants, d'actualité et difficiles pour les organisations. Conséquemment, les gestionnaires sont toujours à l'affût de meilleures solutions pour gérer la performance au sein de leurs organisations.

Une conséquence importante de ne pas avoir de cadre conceptuel de gestion de la performance (Performance Management Framework ou PMF) en place est l'incapacité de différencier le succès de l'échec au sein d'une organisation. Les cadres conceptuels de gestion de la performance sont nécessaires aux organisations qui doivent planifier, assurer un suivi et contrôler leurs activités, ainsi que prendre des décisions éclairées. L'utilisation d'un cadre conceptuel de gestion de la performance peut offrir à une organisation une meilleure vision de son fonctionnement réel et indiquer si elle est en voie d'atteindre ses objectifs.

Au fil des ans, plusieurs cadres ont été développés pour les gérer les actifs tangibles et intangibles de l'organisation. Dans le passé, la gestion de la performance a surtout été orientée vers le point de vue économique. Kaplan et Norton ont ajouté trois autres points de vue dans leur cadre, soit le Balanced Scorecard (BSC), et cet ajout représente une contribution majeure au domaine.

Les cadres de gestion de la performance existants ne satisfont pas aux exigences de la gestion du génie logiciel étant donné que différents points de vue doivent être pris en compte en même temps. De plus, les données quantitatives sous-jacentes sont multidimensionnelles et les techniques de visualisation à deux et trois dimensions ne sont pas adéquates. Troisièmement, chaque organisation a ses propres points de vue de performance qui lui sont spécifiques. Dernièrement, ces points de vue doivent être représentés de façon consolidée pour une saine gestion de l'ensemble.

Le but de cette thèse est de développer un prototype pour la gestion de la performance multidimensionnelle en génie logiciel. La thèse commence par définir les termes importants et les concepts clés utilisés dans la recherche : le logiciel, la performance, la gestion, les modèles multidimensionnels, le développement, l'ingénierie, et le prototype, et les diverses combinaisons de ces termes. Il est suivi par une revue des modèles multidimensionnels de la performance qui sont spécifiques au génie logiciel et des modèles multidimensionnels de performance qui sont disponibles de façon générique en management. Un cadre de gestion de la performance en génie logiciel est proposé qui est divisé en quatre phases : la conception, la mise en œuvre, l'utilisation du cadre, et l'amélioration de la performance. Un prototype est ensuite proposé en appui à ce cadre. Le prototype comprend notamment des outils d'analyse visuelle pour gérer, interpréter et comprendre les résultats sous une forme consolidée tout en permettant l'accès aux valeurs des dimensions individuelles de la performance. De plus, le référentiel de données de projet logiciel mis à disposition par l'International Software Benchmarking Standard Group (ISBSG) est intégré au sein du prototype.

Mots-clés : Modèles multidimensionnels de performance, Mesure de la performance, Gestion de la performance, Visualisation, Management du génie logiciel, International Software Benchmarking Standards Group, ISBSG, QEST, ISO 9126

DEVELOPMENT OF A PROTOTYPE FOR MULTIDIMENSIONAL PERFORMANCE MANAGEMENT IN SOFTWARE ENGINEERING

Vasile STROIAN

ABSTRACT

Managing performance is an important, and difficult, topic, and tools are needed to help organizations manage their performance. Understanding, and improving performance is an important problem.

Performance management has become more and more important for organizations, and managers are always on the lookout for better solutions to manage performance within their organizations.

One of the most important consequences of not having a Performance Management Framework (PMF) in place is the difficulty of differentiating organizational success from failure over time. Performance Management Frameworks have become important to organizations that need to plan, monitor, control, and improve their decisions. Use of a PMF can show an organization how it is performing and indicate whether or not an organization is going in the right direction to achieve its objectives.

Over the years, several frameworks have been developed to address the management of organizational assets, both tangible and intangible. Performance measurement has always mostly been focused on the economic viewpoint. The framework developed by Kaplan and Norton adds three other viewpoints to this, and this addition represents a significant improvement to PMFs.

The PMFs currently proposed do not meet the analytical requirements of software engineering management when various viewpoints must be taken into account concurrently. This difficulty is compounded by the fact that the underlying quantitative data are multi-dimensional, and so the usual two- and three-dimensional approaches to visualization are generally not sufficient to represent such models. Organizations vary considerably in the wide variety of viewpoints that influence their performance, and every organization has their own viewpoints that they want to manage, and which must be represented in a consolidated manner.

The purpose of this thesis is to develop a prototype for managing multidimensional performance in software engineering. The thesis begins by defining the important terms or key concepts used in the research: software, performance, management, model, multidimensional, development, engineering, and prototype, and the various associations of these terms. This is followed by a review of the multidimensional PMFs that are specific to software engineering and the generic multidimensional performance models that are available to management.

A framework for managing performance in software engineering in four phases: design, implementation, use of the framework, and performance improvement is then presented. Based on this framework, a prototype tool is developed. The prototype notably includes visual analytical tools to manage, interpret, and understand the results in a consolidated manner, while at the same time keeping track of the values of the individual dimensions of performance. The repository of software project data made available by the International Software Benchmarking Standard Group (ISBSG) is integrated into and used by the prototype as well.

Keywords : Multidimensional management models, Performance measurement, Performance management, Visualization, Software engineering management, International Software Benchmarking Standards Group, ISBSG, QEST, ISO 9126

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

Page

LIST OF ACRONYMS

APV	Actual Performance Value
APV%	Actual Performance Value after normalization
APV(coef)	Actual Performance Value after taking into account the weight coefficient
BSC	Balanced Scorecard
EFQM	European Foundation for Quality Management
FCM	Factor-Criteria-Metric
FP	Function Point
GQM	Goal-Question-Metric
GE	Guiding Elements
IAM	Intangible Asset Monitor
IEEE	Institute of Electrical and Electronics Engineers
IFPUG	International Function Point Users Group
IPMS	Integrated Performance Measurement System
ISO	International Organization for Standardization
ISBSG	International Software Benchmarking Standards Group
IC	Intellectual Capital
IS	Information Systems
IT	Information Technology
KA	Knowledge Area
QEST	Quality Economic Social Technical
MIM	Measurement Information Model
MSEK	Millions of Swedish Krona
OP_APV	Organizational Actual Performance
OP_TPV	Organizational Target Performance
PDR	Project Delivery Rate
PMI	Project Management Institute
PSM	Practical Software Measurement
PMF	Performance Management Framework

SEI	Software Engineering Institute
SLC	Software Life Cycle
SWEBOK	Software Engineering Body of Knowledge
TPV	Target Performance Value
TPV%	Target Performance Value after normalization
TPV(coef)	Target Performance Value after taking into account the weight coefficient

INTRODUCTION

Purpose of this thesis

The purpose of this thesis is to develop a software prototype for multidimensional performance management in software engineering. The prototype will be primarily based on a geometrical visualization approach adapted from a software multidimensional performance measurement model titled QEST (Quality factor + Economic, Social, and Technical) (Abran and Buglione, 2003; Buglione and Abran, 1999), the Sink and Tuttle viewpoints of organizational performance (Sink and Tuttle, 1989), the ISO 9126 standard approach to classifying software quality characteristics and sub-characteristics (ISO/IEC, 2001), and the International Software Benchmarking Standards Group (ISBSG) repository of software project data (ISBSG, 2008b).

Performance management

In recent decades, performance management has become more and more important for organizations and managers who are always looking for better solutions to manage performance in their organization. Any activity of an organization may influence performance, and organizations need to measure and manage organizational performance in order to plan, monitor, control, and improve their productivity, their efficiency, their effectiveness, the quality of their products, and their ability to deliver them on time.

A successful organization must not only measure performance, but also manage it, and the manner in which performance is managed is becoming an increasingly important research area (Neely, 2005; Neely *et al.*, 1996).

Performance management in the context of software engineering

Managing performance in software engineering is a complex endeavor, requiring tools that take into account the specifics of software development and maintenance. According to The Standish Group's 1995 CHAOS study, only 9% of American IT projects in large organizations were successful, and the two main reasons why these projects failed are the lack of user involvement and the lack of top management support (The Standish Group, 1995). In 2009, of the projects (58% US-based and 24% European) costing more than $750,000 and less than $3 million, only 19% were successful, and "projects over 10 million only have a 2% chance of coming in on time and on budget" (The Standish Group, 2009). Note, however, that an incomplete description of how they conducted this study makes it difficult to evaluate the validity of the research, even though it is very widely cited in the scientific and industrial literature (Eveleens and Verhoef, 2010; Glass, 2005; Jørgensen and Moløkken-Østvold, 2006).

Nevertheless, even in the case of the most optimistic failure rates, which would be 10-15% according to (Glass, 2005), these project failures still represent a colossal amount of money. For example, spending in the U.S. alone on information technology, which represents about 50% of their total corporate capital spending, was estimated at roughly $2.5 trillion for 2005 (Knowledge@Wharton, 2005).

According to Glass, 'success' and 'failure' are tricky words, because "how do you categorize a project that is functionally brilliant but misses its cost or schedule targets by 10 percent?" (Glass, 2005). According to Donaldson and Siegel, it is difficult to say what a 'good' software product or development process is, because it depends on your point of view (Donaldson and Siegel, 2001, p. 9). The main criteria for determining a successful software development project are: was it delivered on time? can it do what it is supposed to do? and, finally, was it delivered for the agreed cost (Donaldson and Siegel, 2001, p. 2; Ewusi-Mensah, 2003).

Unfortunately, the software development industry has long suffered from projects being late, over budget, and lacking in agreed features, owing to technical or time limitations (DeMarco, 1982, pp. 1-2; Ewusi-Mensah, 2003; Taylor, 2000; The Standish Group, 1995).

The management of software development projects in general, and of their performance in particular, are not easy, primarily for the following reasons:

- There is no unique way to arrive at results in software development: the organization "must set up a way of doing business that allows for adaptation to the situation at hand…no 'way' of doing business can anticipate all contingencies" (Donaldson and Siegel, 2001, p. 5).
- Software is often built using an iterative process, rather than by following a predefined sequence of closed tasks. In a more traditional engineering field, such as civil engineering, we would build the first floor of an apartment building, and then, when that is finished, build the next floor, and so on. This is often not the case for software development (Vliet, 2008, pp. 8-9).
- There is no simple explanation for failure in software development projects, and the factors that contribute to failure are multidimensional (Ewusi-Mensah, 2003).
- Creativity and discipline are necessary for software development, and these are difficult to combine. Discipline is necessary for achieving predictability, while creativity is a must if we wish to produce innovative results (Vliet, 2008, p. 7). According to Vliet, "discipline is one of the keys to the successful completion of a software development project" (Vliet, 2008, p. 7).
- There are few underlying theories in software engineering, and its fundamental principles are not well established, compared to those of more traditional fields of engineering (Bourque *et al.*, 2002; Meridji, 2010). We must always remember that computing in general is only roughly sixty years old, and that the word 'software' was only coined in 1958 (Shapiro, 2000). In terms of software engineering specifically, the Guide to the Software Engineering Body of Knowledge has only been in existence now for six years (Abran *et al.*, 2004).
- The degree of novelty is extremely high in software engineering: we are faced with an extremely rapid rate of change in the underlying technology. "Software development involves novelty, which introduces uncertainty. It can be argued that there is a higher degree of novelty in software than in other forms engineering." (Laplante, 2004, p. 167).

What does 'multidimensional' mean in the context of performance management?

Managing performance is challenging because of its multidimensional nature, its many levels of granularity, and the fact that organizations differ so widely.

Historically, performance measurement has always been focused on economical or financial concerns, meaning that performance management systems "were uni-dimensional – focused purely on financial measures." (Bourne *et al.*, 2003).

According to Simons (Simons, 1995, p. 13), performance measurements by themselves are obsolete, because they are backward-looking and not forward-looking. He also believes that organizations should be analyzed multidimensionally, because "…organizations are also sets of relationships among self-interested participants, each of whom is balancing personal well-being and organizational needs."

Performance measurement "is complicated by its multi-dimensional nature...managers need to resolve issues such as: conflicts between performance measures; the appropriate balance of internal and external measures; the linking of measures and strategy; etc."(Neely *et al.*, 1996).

Every organization has their viewpoints (dimensions) from which they wish to evaluate the performance of their organization. The most common and widely analyzed viewpoint in software engineering has always been the economic one (managers' viewpoint), but the social or socio-organizational viewpoint (users' viewpoint) and the technical or socio-technical viewpoint (for example, the software engineers' viewpoint) also have an important impact on performance (Buglione and Abran, 1999; Ewusi-Mensah, 2003):

- The socio-organizational viewpoint can be characterized as "where the organizational component describes or deals with the set of organizational interests the artifact must serve" (Ewusi-Mensah, 2003);

- The socio-technical viewpoint mainly concerns the requirements of the software and the necessary supporting documents;
- The economic viewpoint reflects the cost of creating the software artifact.

Initially, performance measurement focused only on financial measures, but organizations also need non financial measures in order to better align objectives and strategies throughout all levels. Non financial measures, however, are not easy to quantify.

The main objective of a Performance Management Framework (PMF) is to help an organization move in the desired direction. Performance management is a complex task and cannot be managed with only a single indicator (Sink and Tuttle, 1989, p. 59). Many multidimensional PMFs have been developed and proposed in the literature to help organizations manage their organizational performance (Neely, 2005).

For example, the concept of the *Tableau de bord* has been used in the francophone world to manage performance for almost 80 years now. Briefly, this is a dashboard developed for top management to monitor the operations of an organization. However, there is no standard *Tableau de bord*, and almost every author proposes a different one. It usually includes financial and non financial measures; however there is little linkage with the strategy of an organization (Balantzian, 2005).

In the last three decades or so, proposals for performance frameworks have grown in number. Among the better known frameworks are the Balanced Scorecard (Kaplan and Norton, 1992; 1993; 1996a; 1996b), the Performance PRISM (Neely and Adams, 2003; Neely *et al.*, 2002), the EFQM (European Foundation for Quality Management) Excellence Model (European Foundation for Quality Management, 2008), and the Baldrige Criteria for Performance Excellence (Baldrige, 2010; Srivivatanakul and Kleiner, 1996; Tudahl and Lindner, 1994).

The Balanced Scorecard (BSC), originally published in 1992, is perhaps the best-known PMF and the one most often used in the U.S. It identifies four viewpoints from which to manage organizational performance: financial, internal process, external customer, and innovation and learning (Kaplan and Norton, 1992; 1993; 1996a; 1996b).

According to (Kaplan and Norton, 1992), an important advantage of using the BSC is that it helps managers focus on strategy and vision to manage organizational performance. The use of these four viewpoints, which combine financial and non financial measures, can "help managers transcend traditional notions about functional barriers and ultimately lead to improved decision making and problem solving."

According to Marr and Neely (Marr and Neely, 2003, p. 7), more than 50% of the largest U.S. firms had adopted the BSC by the end of 2000, and 43% of the companies that were not using it were planning to do so. However, it cannot be applied directly to any organization, because they are all different and require particular viewpoints and measures associated with their own objectives (Kaplan and Norton, 1993). The Balanced IT Scorecard (BITS) (Reo *et al.*, 1999) is a specific version of the four original viewpoints for the information technology industry, with the addition of a fifth, the 'people' viewpoint.

A PMF can be very complex, primarily because of its multidimensionality, and no single approach is suitable for all organizations. There is no 'one size fits all' PMF. However, there seems to be a consensus in the literature that the objectives of a PMF amount to more than simply monitoring measures. In addition, performance has multiple perspectives or viewpoints. These viewpoints and measures also need to be combined in some way into a single, overall performance value for management purposes.

The Performance Prism takes into consideration the complexity of performance management by using five interlinked points of view and representing them in the geometrical form of a prism with five facets (Neely and Adams, 2003; 2005b). The QEST model (Abran and Buglione, 2003) proposes a performance value that is, essentially, a normalized measurement value, and combines the various views of performance measurement. The geometrical representation adopted by QEST-3D is a regular tetrahedron with a regular triangular base and equal sides (a pyramid), which is suitable for three viewpoints only. However, where there are more than three viewpoints, no visual representation is possible. The QEST model can also be used for consolidating BSCs (Abran and Buglione, 2003).

According to Kaplan and Norton, “managers recognize the impact that measures have on performance. But they rarely think of measurement as an essential part of their strategy” (Kaplan and Norton, 1993). Some keys to better management of organizational performance are a clear definition of objectives and paying attention to the selection and weighting of measures and viewpoints.

From one-dimensional to multidimensional performance models

The economic viewpoint has always been the main focus in evaluating organizational performance. The best-known method for measuring performance at the beginning of 20th century was probably the DuPont Formula, which originated before 1920: performance measurement is expressed by the relationships between five economic indicators (operating profit margin, asset turnover, financial leverage, cost of debt, and tax retention rate), each resulting from calculations based on two ratios: return on investment (ROI) and return on equity (ROE) (12MANAGE, 2008; Posey, 2006; Ramesh, 2000, p. 66). The ROI is obtained by dividing net profit by total assets. The ROE ratio, often called the DuPont Formula, has the following equation:

$$ROE = \frac{\text{Net profit}}{\text{Equity}} = \frac{\text{Net profit}}{\text{Pretax profit}} * \frac{\text{Pretax profit}}{\text{EBIT}} * \frac{\text{EBIT}}{\text{Sales}} * \frac{\text{Sales}}{\text{Assets}} * \frac{\text{Assets}}{\text{Equity}}$$

where:
Net profit = Profit after taxes;
Equity = Shareholders' equity;
EBIT = Earnings before interest and taxes;
Sales = Net sales.

One of the problems with performance measurement initially was that too much importance was given to the economic viewpoint (Bourne *et al.*, 2003), as clearly exemplified by the Dupont formula.

Managers "have tracked quality, market share, and other non-financial measures for years. Tracking these measures is one thing. But giving them equal (or even greater) status in determining strategy, promotions, bonuses, and other rewards is another" (Eccles, 1991). Performance management based only on an accounting viewpoint "often fails to support investments in new technologies and markets that are essential for successful performance in global markets" (Eccles, 1991).

PMFs have been developed to achieve a more balanced view between internal and external viewpoints, and between economic and non economic measures and indicators. Kaplan and Norton identified the weakness of early models, which was that they were one-dimensional, and added three other dimensions that are relatively easy to understand and can be applied to any organization (Kaplan and Norton, 1992). Their BSC balances the economic viewpoint by adding those three new viewpoints, which are outside the economic or financial sphere (Kaplan and Norton, 1992; 1993; 1996a; 1996b)

According to Kaplan and Norton, there are organizations using fewer than four viewpoints and others that need additional viewpoints. They find no theoretical support for the statement that four viewpoints are necessary and sufficient (Kaplan and Norton, 1996a, p. 34). Organizations vary considerably with regard to the wide array of dimensions that influence their performance. Generally, however, performance is multidimensional, and there is a need to measure all the viewpoints that have an important impact on organizational success.

Visualization and consolidation techniques are necessary for managing multidimensional performance data in the software engineering field

An organization is multidimensional, and overall performance should be managed with the support of appropriate graphical tools that are able to represent this multidimensionality. Analyzing multidimensional organizational performance, and exploring performance data, can be difficult without proper visualization tools or approaches (Lawton, 2009) .

Having the right information at the right time is crucial to making the right decisions, and so there is growing interest in data visualization in all disciplines, including engineering and management (Stroian *et al.*, 2006).

One difficulty with current performance models is to represent many possible viewpoints quantitatively and in a consolidated manner, while at the same time keeping track of the values of the individual dimensions of performance (Abran and Buglione, 2003; Bourque *et al.*, 2006; Stroian *et al.*, 2006). This difficulty is compounded by the fact that the underlying quantitative data are of high dimensionality, which means that the usual two- and three-dimensional approaches to visualization are not sufficient for representing such models.

QEST model and prototype

The QEST (Quality factor + Economic, Social, and Technical dimensions) model represents performance in a consolidated manner and is capable of handling independent sets of viewpoints (dimensions) without predefined ratios and weights. This model produces a single performance value, which is important to good decision making because it integrates individual and linked measurements into a single performance indicator, making it possible to see the ‘big picture’ and not always getting caught up in the details.

It adopts a geometrical approach (a regular tetrahedron representation is the basis for this 3D-model) (Abran and Buglione, 2003) to representing performance as is shown in Figure 2.9. A tetrahedron is composed of four triangular faces, three of which meet at a single point (each vertex represents a viewpoint), and a regular tetrahedron is one consisting of four regular triangles (triangular pyramid). The concepts of distance, area, and volume are used to represent performance.

At the beginning of this research project, a prototype was available to support the QEST approach; however, this prototype (Abran *et al.*, 2003a):

- is limited with regard to software product quality, as its measures are restricted to those proposed in the context of the ISO 9126 quality characteristics and sub-characteristics (ISO/IEC, 2001);
- is constrained to three predefined viewpoints: social, technical, and economic;
- does not include historical data and has no mechanisms for handling such data;
- does not include a repository for persistent data storage;
- is limited to a pyramidal representation of data.

ISO 9126

The International Organization for Standardization introduced a standard named ISO/IEC 9126 (1991) to interpret and measure software quality (ISO/IEC, 2001). This standard defines a software quality model using:

- three viewpoints, defined as follows: quality in use, external quality, and internal quality;
- six characteristics: functionality, system reliability, usability, efficiency, maintainability, and portability. Two or more sub-characteristics are associated with each characteristic.

Sink and Tuttle model

The Sink and Tuttle model can be applied to improve organizational performance. It provides operational definitions of seven performance criteria: effectiveness, efficiency, quality, productivity, quality of work life, innovation, and profitability (Sink, 1985; Sink and Tuttle, 1989). These authors insist on the importance of measurement system design in improving organizational performance and achieving an accepted vision regarding performance improvement. They claim that organizations have the responsibility of getting the job done on time, within quality specifications, and with the right amount of resources, and to continuously improve individual, group, organizational, and performance systems.

The ISBSG repository as a benchmarking and performance tool specific to software engineering

Benchmarking is used to compare organizational performance results with those of other organizations. The International Software Benchmarking Standards Group (ISBSG), a not-for-profit organization established in 1994, has established and maintains a database of software project data that can be used by software project managers for various purposes, including estimation and benchmarking (ISBSG, 2008b).

ISBSG offers an interesting avenue for managing performance quantitatively:

- There are more than 100 nominative and quantitative variables included in the repository;
- The ISBSG Repository, release 11 (R11), made available in 2009, contains data on 5,000 projects. Data from a large number and a wide variety of projects are therefore available;
- The repository can be used as a valuable alternative for organizations without their own performance database. If they do have their own performance database, then the ISBSG repository can be used as an external supplement to that internal one.

Problem statement of this thesis

The purpose of this thesis is to develop a prototype to model multidimensional performance in software engineering management.

The generic problem addressed is how to better manage performance in software engineering when many dimensions or viewpoints must be taken into account concurrently:

- The PMFs currently proposed do not meet the analytical requirements of software engineering management when various viewpoints must be taken into account concurrently;

- There is a lack of visualization techniques and analysis tools when multi-dimensionality must be taken into account;
- There already exist a significant number of models in software engineering to manage quality, but there are few PMFs available that are specific to software engineering (Bourque *et al.*, 2006; Stroian *et al.*, 2006);
- There is currently no satisfactory tool available to meet the analytical requirements of software engineering management when various viewpoints must be taken into account individually, but at the same time concurrently.

Research questions

From the above problem statements, the following research questions have been formulated:

- Question 1: A. What are the strengths and weaknesses of the multidimensional PMFs currently available in software engineering, and in management more generally?
  B. Which of the multidimensional PMFs could be used for performance management in software engineering?

- Question 2: What are the strengths and weaknesses of the QEST prototype?

- Question 3: How can we build a prototype for multidimensional performance management in software engineering to represent, graphically and in a consolidated manner, the many possible performance viewpoints, while at the same time keep track of the values of the individual performance dimensions?

Research objectives

The main objective of the research is to develop a prototype for multidimensional performance management in software engineering which can represent, quantitatively and in consolidated manner, the many possible performance viewpoints, while at the same time keeping track of the values of the individual performance dimensions (Bourque *et al.*, 2006; Stroian *et al.*, 2006).

Research phases and thesis outline

The research phases leading to this thesis are as shown in Figure 0.1. In the first phase, the main or key concepts of this thesis are defined and explained, the strengths and weaknesses of the models and frameworks found in the literature are analyzed, the repository of the International Software Benchmarking Standard Group (ISBSG) is presented, and, finally, the QEST prototype currently available is analyzed for its strengths and weaknesses. The main purpose of this phase is to analyze the models and frameworks and to identify candidate concepts to be included in the conceptual framework.

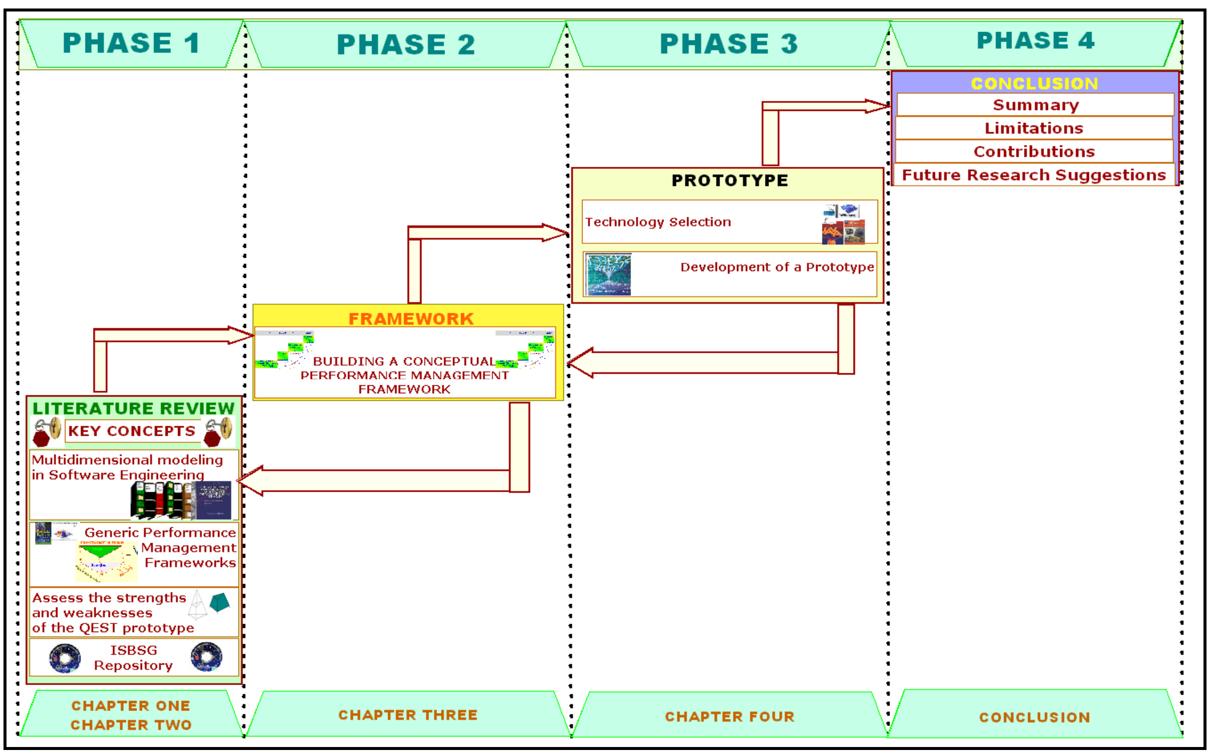


Figure 0.1 Research phases and thesis outline.

In the second phase, a conceptual framework for managing multidimensional performance in software engineering is proposed. This conceptual framework is composed of four main steps:

- design: decide what performance to measure and how to measure it;
- implementation of measures: collect the measures;
- use analysis and visualization techniques;
- improve performance.

An effective PMF incorporates a multidimensional view of performance, provides feedback, and enables performance improvement. This conceptual framework is a major input to the prototype discussed next.

The third phase focuses on the primary goal of the dissertation, which is to develop a prototype for multidimensional performance management in software engineering. The technology is selected and a prototype called MultiPERF is developed to manage multidimensional performance in software engineering.

Finally, a summary of the research, the contribution and limitations of the research, and future research suggestions are presented in the fourth and final phase.

The overall structure of the thesis is illustrated in the lower part of Figure 0.1. The dissertation is structured in five chapters.

The results of Phase 1 are discussed in chapters 1 and 2. Chapter 1 defines the important terms or key concepts used in this thesis: software, performance, management, model, multidimensional, development, software, engineering, and prototype, and the various associations of these terms.

Chapter 2 includes:

- a review of the multidimensional PMFs that are specific to software engineering;

- a review of the generic multidimensional performance models that are available to management;
- a review of the repository of the International Software Benchmarking Standard Group (ISBSG);
- an analysis of the strengths and weaknesses of the QEST prototype.

Chapter 3 describes a conceptual framework for managing performance in software engineering. It comprises four phases:

- Phase 1: design – decide what to measure and how to measure it;
- Phase 2: implementation – collect the measures;
- Phase 3: use of the framework – visualization, analysis, and interpretation of the results;
- Phase 4: performance improvement – provide feedback and facilitate benchmarking.

Chapter 4 presents and discusses the prototype that was developed in this thesis and which is based on the conceptual framework discussed in chapter 3:

- the features of the prototype;
- a transparent simulation approach to fixing appropriate performance targets using the ISBSG repository;
- a discussion of how the prototype enables or supports the framework proposed in the previous chapter;
- relation of the results to the research questions.

Conclusions summarizes the main results of the thesis, discusses their originality and their limitations and how they contribute to the advancement of knowledge, and offers suggestions for future work within this research theme.

Contributions of the research

The research contributions of this dissertation are the following:

- Framework: the proposal of a conceptual framework for managing organizational performance in the context of software engineering;
- Prototype: the development of a prototype for managing performance in the context of software engineering which:
  - supports the conceptual framework mentioned above;
  - adopts a multidimensional geometrical approach adapted from the QEST model to facilitate the management of performance when many dimensions or viewpoints must be taken into account individually and concurrently (Abran and Buglione, 2003; Buglione and Abran, 1999).
  - facilitates the use of the ISO 9126 quality standard (ISO/IEC, 2001) in a performance management context;
  - adopts the Sink and Tuttle viewpoints of organizational performance (Sink, 1985; Sink and Tuttle, 1989);
  - incorporates a repository of software project data known as the ISBSG repository (ISBSG, 2008b).

These contributions have been published by/accepted for the following conferences:

- Stroian, Vasile, Pierre Bourque, and Alain Abran. 2006. "Visualization – a key concept for multidimensional performance modeling in software engineering management." IEEE-TTTC International Conference on Automation, Quality & Testing, Robotics AQTR 2006 (THETA 15), (May 25-28, 2006), Cluj-Romania, p. 6.

- Bourque, Pierre, Vasile Stroian, and Alain Abran. 2006. "Proposed concepts for a tool for multidimensional performance modeling in software engineering management." IEEE-IES International Symposium on Industrial Electronics (ISIE), (July 9-13, 2006), Montreal-Canada, p. 6.

CHAPTER 1

PRESENTATION OF KEY CONCEPTS

As shown in Figure 1.1 in the Phase 1 column, this chapter and the following one present a review of the literature on the multidimensional PMFs. The objective of chapter 1 is to explain the key concepts of this research, as shown in Figure 1.2. Chapter 2 will present and analyze the multidimensional PMFs that are currently available in software engineering specifically, and in management in general, assess the strengths and weaknesses of the QEST prototype, and present the ISBSG repository.

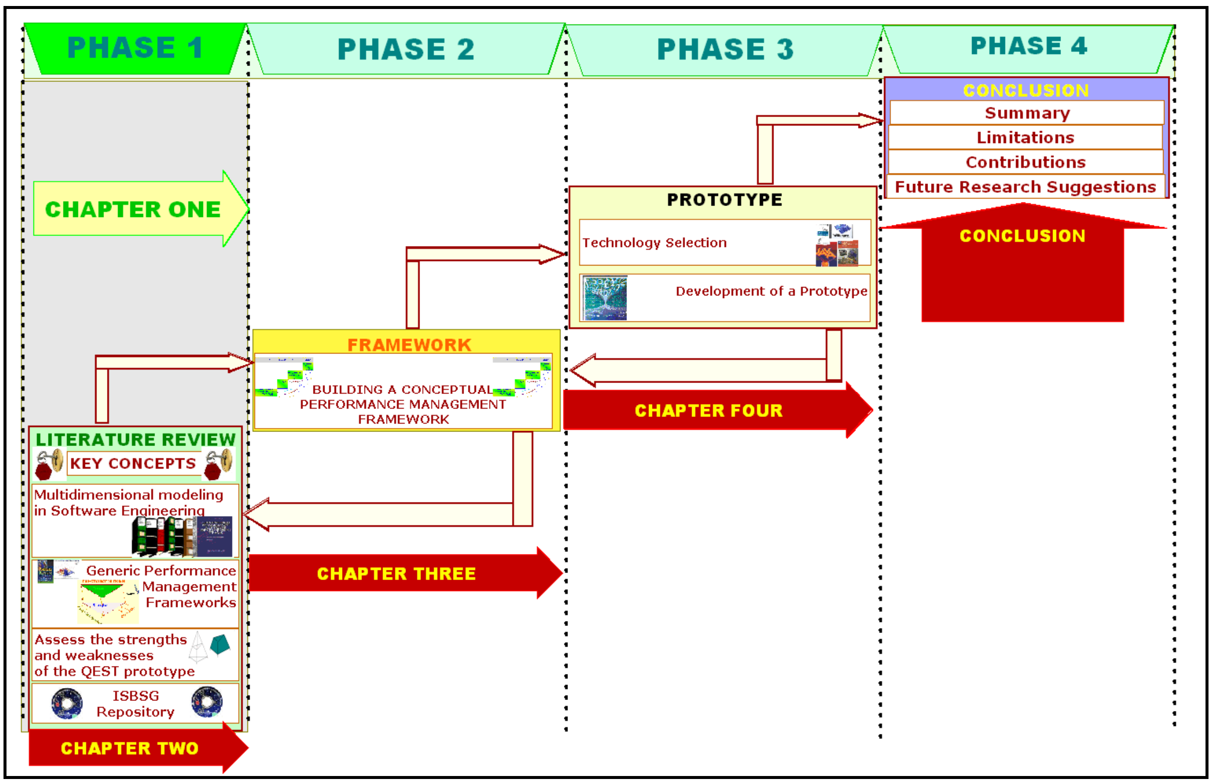


Figure 1.1 Thesis – literature review – Phase 1.

The key concepts discussed in this chapter all revolve around the title of the thesis. This is why the top box (green background color) in Figure 1.2 contains the title of the thesis. The second-level boxes contain words extracted from the title: development, prototype, multidimensional, performance, software, engineering, and management. Each box has a uniquely colored arrow to link the box with the next, or third, level. The bottom level boxes (white background color) indicate the sections in this chapter where these key concepts are defined and explained.

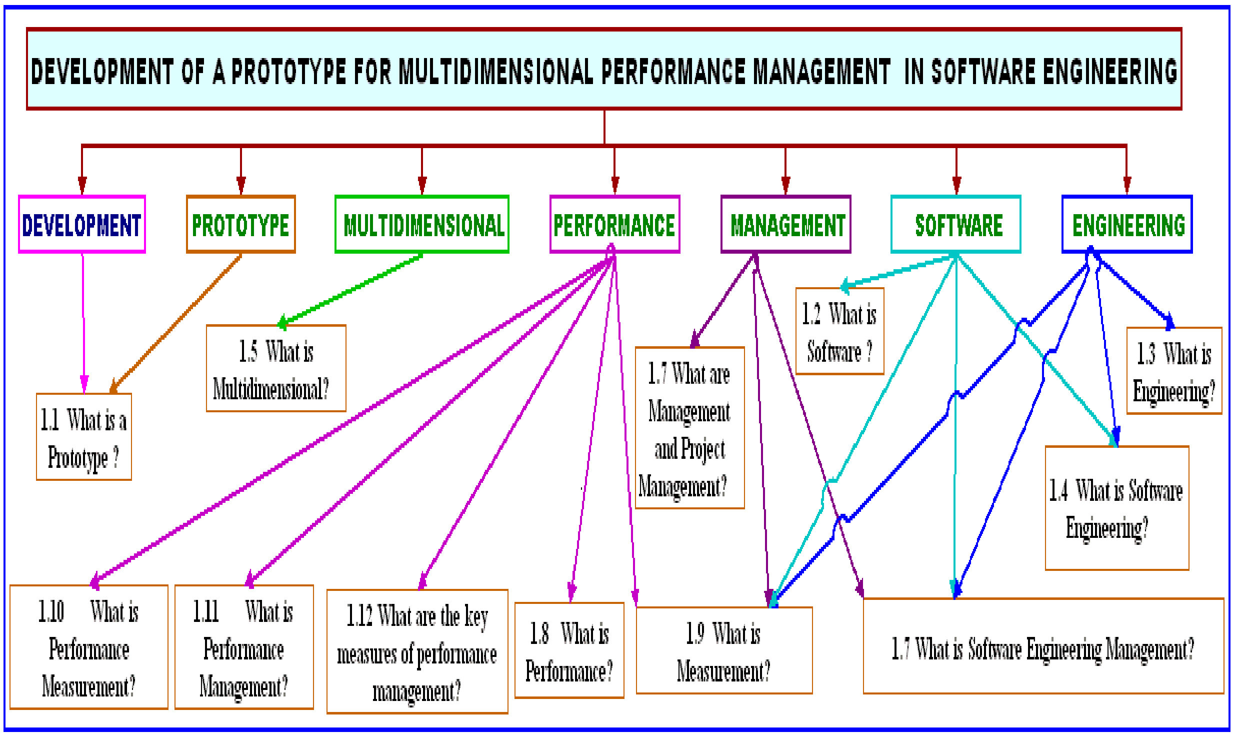


Figure 1.2 Key concepts – related to the thesis title.

1.1 What is a prototype?

In Webster's Dictionary (Merriam-Webster, 2005) a prototype is defined as 'an original model on which something is patterned.' According to McGraw-Hill's Encyclopedia of Science & Technology, a prototype is defined as 'a first or original model of hardware or software.' (McGraw-Hill's, 2008).

Experimental prototyping "is used for the investigation of alternative approaches to a solution", and exploratory prototyping "is used to assist in the formulation of a problem or issue" (McGraw-Hill's, 2008). The prototype developed in this thesis belongs much more in the exploratory category than in the experimental category.

One advantage of using prototyping is "the ability to rapidly construct a product that is representative of the final product at relatively low cost and effort" (McGraw-Hill's, 2008). But, "a prototype is generally a functionally immature model of a proposed product that is built to explore requirements, investigate alternative approaches, or demonstrate model feasibility. The model may or may not evolve into a mature, functionally useful product" (McGraw-Hill's, 2008).

In the context of software engineering, the Guide to the Software Engineering Body of Knowledge (SWEBOK) (Abran *et al.*, 2004) defines a prototype as:

> "a means for validating the software engineer's interpretation of the software requirements, as well as for eliciting new requirements. As with elicitation, there is a range of prototyping techniques and a number of points in the process where prototype validation may be appropriate. The advantage of prototypes is that they can make it easier to interpret the software engineer's assumptions[,] and, where needed, give useful feedback on why they are wrong."

1.2 What is software?

One definition of software is "the programs, routines, and symbolic languages that control the functioning of the hardware and direct its operation" (HMC, 2000).

Software is defined by the IEEE as "computer programs, procedures and possibly associated documentation and data pertaining to the operation of a computer system" (IEEE Computer Society, 1990).

The problem of loosely deliverables, notably in the IEEE definition, is why Pressman explicitly includes data structures and documentation within the scope of software:

> "Software is (1) instructions (computer programs) that when executed provide desired function and performance, (2) data structures that enable the programs to adequately manipulate information, and (3) documents that describe the operations and use of the programs." (Pressman, 2001, p. 6)

1.3 What is engineering?

According to Vincenti, "engineers spend their time dealing mostly with practical problems" (Vincenti, 1990, p. 200), and, in his book on the development of aeronautical engineering, he argues strongly that, from the point of view of the practitioner, "technology, though it may apply science, is not the same as or [*sic*] entirely applied science" (Vincenti, 1990, p. 4).

The McGraw-Hill Encyclopedia of Science and Technology defines engineering as "the science by which the properties of matter and the sources of power in nature are made useful to humans in structures, machines, and products" (McGraw-Hill's, 2008).

The IEEE definition of engineering is the "application of a systematic, disciplined, quantifiable approach to structures, machines, products, systems or processes" (IEEE Computer Society, 1990).

1.4 What is software engineering?

The IEEE definition of software engineering is an adaptation of their definition of engineering to the specifics of software (IEEE Computer Society, 1990): "(1) The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software" and "(2) The study of approaches as in (1).".

According to Pressman, "Software engineering encompasses a process, the management of activities, technical methods, and tools" (Pressman, 2001, p. 21), and software is developed by applying three distinct phases (Pressman, 2001, p. 22):

- the definition phase, focusing on "what information is to be processed, what function and performance are desired, what system…, what interfaces…, what design…, and what validation criteria are required to define a successful system" (Pressman, 2001, p. 22)
- the development phase, focusing on "how data are to be structured, how function is to be implemented…, how procedural details…, how interfaces..., how the design will be translated into a programming language…, and how testing…" (Pressman, 2001, p. 22)
- the support phase, focusing on changes "associated with error correction, adaptations required…, and changes due to enhancements brought about by changing customer requirements…reapplies the steps of the definition and development phases, but does so in the context of existing software" (Pressman, 2001, p. 22).

Software engineering, according to Berry, is:

> "that form of engineering that applies a systematic, disciplined, quantifiable approach, the principles of computer science, design, engineering, management, mathematics, psychology, sociology and other disciplines as necessary and sometimes just plain invention, to creating, developing, operating and maintaining cost-effective, reliably correct, high-quality solutions to software problems" (Berry, 1992).

The IEEE Computer Society carried out a project to develop a Guide to the Software Engineering Body of Knowledge (SWEBOK) to establish consensus on generally accepted knowledge in the Software Engineering discipline. The SWEBOK objectives are:

> "(1)'To promote a consistent view of software engineering worldwide; (2) To clarify the place–and set the boundary–of software engineering with respect to other disciplines such as computer science, project management, computer engineering, and mathematics; (3) To characterize the contents of the software engineering discipline; (4) To provide a topical access

> to the Software Engineering Body of Knowledge; (5) To provide a foundation for curriculum development and for individual certification and licensing material" (Abran *et al.*, 2004).

The ten Knowledge Areas (KAs) identified by this initiative as specific to software engineering are: Software requirements, Software design, Software construction, Software testing, Software maintenance, Software configuration management, Software engineering management, Software engineering process, Software engineering tools and methods, and Software quality (Abran *et al.*, 2004).

1.5 What is multidimensional?

In the Merriam-Webster Collegiate Dictionary (Merriam-Webster, 2005), dimension is notably defined as a 'measure in one direction', 'one of a group of properties whose number is necessary and sufficient to determine uniquely each element of a system of usually mathematical entities (as an aggregate of points in real or abstract space)', or 'the number of elements in a basis of a vector space'.

In the Word Reference (WordReference, 2005) dictionary, 'multidimensional' is defined as 'having or involving or marked by several dimensions or aspects'.

1.6 What are management and project management?

In the Merriam Webster Dictionary again (Merriam-Webster, 2005), management is defined as 'the act or art of managing: the conducting or supervising of something', or, for an organization, it can be defined as 'the collective body of those who manage or direct an enterprise'.

The Project Management Institute (PMI) has defined project management as "the application of knowledge, skills, tools, and techniques to project activities to meet project requirements" (PMI, 2000, p. 6).

According to Kerzner, a decade ago, "project management resided only in the project-driven sectors of the marketplace. In these sectors, the project managers were given the responsibility for profit and loss, which virtually forced companies to treat project management as a profession" (Kerzner, 2009, p. 50). In the last decade, the acceptance of project management in the non-project-driven and hybrid sectors led to "project management being promoted by marketing, engineering, and production, rather than only by the project-driven departments" (Kerzner, 2009, p. 50).

1.7 What is software engineering management?

Software Engineering Management is defined in (IEEE Computer Society, 1990) as "the application of management activities—planning, coordinating, measuring, monitoring, controlling, and reporting—to ensure that the development and maintenance of software is systematic, disciplined, and quantified."

The Software Engineering Management KA of the SWEBOK Guide (Abran *et al.*, 2004) therefore addresses the management and measurement of software engineering. The SWEBOK Guide identifies six major sub areas within the Software Engineering Management KA:

> "Initiation and scope definition, […] the decision to initiate a software engineering project; Software project planning, […] the activities undertaken to prepare for successful software engineering from a management perspective; Software project enactment, […] software engineering management activities that occur during software engineering; Review and evaluation, […] assurance that the software is satisfactory; Closure,…the post-completion activities[…]; Software engineering

> measurement,…the effective development and implementation of measurement programs […]” (Abran *et al.*, 2004).

1.8 What is performance?

According to Lebas, it is difficult to define performance, and few people agree on what performance means, because “it can mean anything from efficiency, to robustness or resistance or return on investment, or plenty of other definitions never fully specified.” (Lebas, 1995).

According to Robbins, the performance of an individual is defined as:

> “a function (f) of the interaction of ability (A) and motivation (M); that is, performance = f(A x M) [*recte* performance = *f(A,M)*]. If either ability or motivation is inadequate, performance will be negatively affected. This helps to explain, for instance, the hardworking athlete with modest abilities who consistently outperforms her more gifted, but lazy, rivals. But an important piece […] is still missing. We need to add opportunity (O) […] Performance = f(A x M x O) [*recte* performance = *f(A,M,O)*]. Even though an individual may be willing and able, there may be obstacles that constrain performance.”(Robbins, 2007, p. 70)

According to the Encarta Dictionary, performance is associated with working effectiveness and represents “the way in which somebody does a job, judged by its effectiveness” (Encarta, 2008). A performance appraisal represents “the assessment of an employee's effectiveness, usually undertaken at regular intervals” (Encarta, 2008).

Some of the definitions in the Webster Dictionary (Merriam-Webster, 2005) for performance are: ‘the execution of an action’, ’something accomplished’, ‘the fulfillment of a claim, promise, or request’, ‘the ability to perform : efficiency’, and ‘the manner in which a mechanism performs’.

According to the Business Dictionary (BusinessDictionary, 2005), performance is defined as the ‘accomplishment of a given task measured against preset standards of accuracy, completeness, cost, and speed’.

In the case of software performance (not organizational performance), it can be defined as the capability of a system to process a given number of tasks in a predetermined time interval (Schmietendorf *et al.*, 2000).

Lebas defines performance as:

> “deploying and managing well the components of the causal model(s) that lead to the timely attainment of stated objectives within constraints specific to the firm and to the situation. Performance is therefore case specific and decision-maker specific. Achieving congruence as to the definition of the parameters of performance and the causal model(s) that lead to it is one of the essential functions of management” (Lebas, 1995).

There are a variety of ways to view and define performance, as illustrated above. However, it is important to understand that this thesis deals with organizational performance, not software performance, individual performance, or any other type of performance. The author’s outlook on organizational performance corresponds to the last definition of performance in this section, as offered by Lebas.

1.9 What is measurement?

Metrology is the scientific study of measurement, and its terminology is standardized in a document titled “International Vocabulary of Basic and General Terms in Metrology” (ISO, 1993). The term ‘metrology’ ‘includes all aspects of measurement (theoretical and practical), collectively referred to in the metrology literature as ‘the science of measurement’ (Abran *et al.*, 2003b), as shown in Figure 1.3, where:

- Metrology is the science of measurement;

- The principles of measurement form the scientific basis for a measurement;
- A method of measurement is the logical sequence of operations for quantifying an attribute. A subjective method of measurement involves human judgment, while an objective measurement method is based on numerical rules;
- Measurement is the implementation of a set of operations for quantifying an attribute using a specified scale. The output is represented by the measurement results (the term 'input' is not defined in the ISO standard) (Abran *et al.*, 2003b).

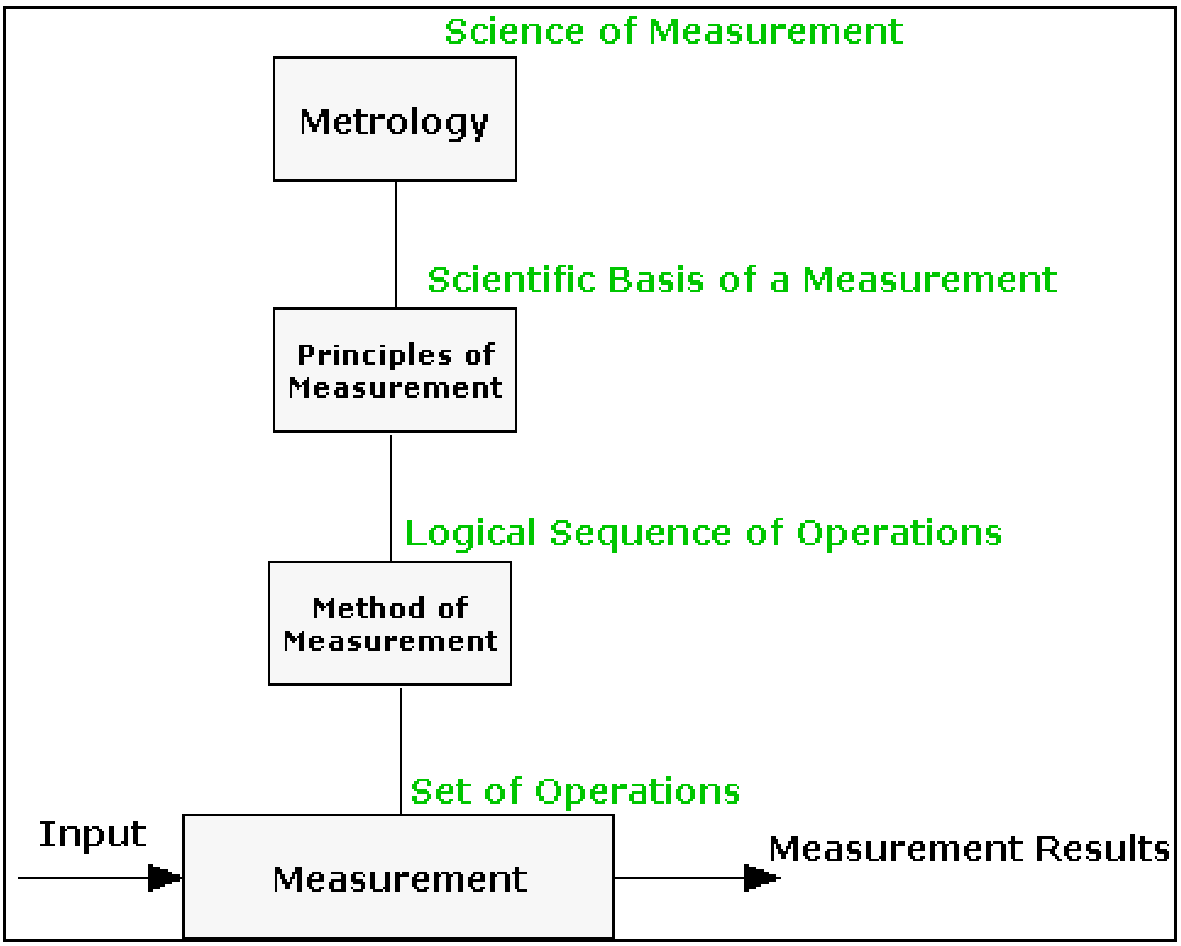


Figure 1.3 Measurement foundations.
Adapted from Abran *et al* (2003b, p. 6)
Reprinted with the permission of the author

The field of metrology covers three main activities: "the definition of internationally accepted units of measurement…the realisation of units of measurement by scientific methods…the establishment of traceability chains by determining and documenting the value and accuracy…and disseminating that knowledge" (Howarth and Redgrave, 2008, p. 9).

According to Vincenti in his book on the emergence of the aeronautical engineering discipline, engineering design knowledge can be divided into six categories:

- Fundamental design concepts, which fall into two important groups:
    - operational principles – as explained by Polanyi, "how [the design's] characteristic parts…fulfil their special function in combining to [*sic: into*] an overall operation which achieves the purpose" (Vincenti, 1990, p. 208);
    - normal configurations – "general shape and arrangement that are commonly agreed to best embody the operational principle" (Vincenti, 1990, p. 209);
- Criteria and specifications – "Engineers, to carry out their task of designing devices, must work to very concrete objectives. This requires that they devise relevant design criteria and specifications" (Vincenti, 1990, p. 213);
- Theoretical tools – "intellectual concepts for thinking about design" (Vincenti, 1990, p. 213) and "mathematical methods and theories for making design calculations"(Vincenti, 1990, p. 213);
- Quantitative data:
    - descriptive – "knowledge of how things are. Descriptive data needed by designers include physical constants…as well as properties of substance...and of physical processes…" (Vincenti, 1990, p. 216);
    - prescriptive – "knowledge of how things should be to attain a desired end" (Vincenti, 1990, p. 217);
- Practical considerations – "designers also need for their work an array of less sharply defined considerations derived from experience in practice" (Vincenti, 1990, p. 217). These considerations are mostly learned on the job;
- Design instrumentalities – "the procedures, ways of thinking and judgemental skills" (Vincenti, 1990, p. 219) necessary to carry out engineering design.

The above categorization of engineering design knowledge is very interesting, notably because it clearly emphasizes the importance of measurement (quantitative data) to the engineering discipline in general.

Bunge states that any quantitative observation is a measurement, and "in order to decide what kind of measurement is to be done, an analysis of the concept denoting the corresponding property must be performed. Accordingly, the nature of quantification must be analyzed before the features of measurement can be understood" (Bunge, 1967, p. 194).

Measurement cannot guarantee success, but it can help an organization manage performance and have a transparent approach to improving performance. In the opinion of Donaldson and Siegel, "measurement for measurement's sake is a waste of time and money. Measurements need to be expressed in everyday terms that are familiar to the organization; otherwise, they may be of little value" (Donaldson and Siegel, 1998). Measuring is applied to past actions, and, according to DeMarco, "you can't control what you can't measure" (DeMarco, 1982, p. 1).

According to Sink and Tuttle, the decision to measure is imposed on us "at least in the control-oriented situation. It is frequently avoided, and almost always misunderstood certainly influenced by management style and preference…motivated by unnecessary or unwarranted desires to control, or at least created illusion of control" (Sink and Tuttle, 1989, p. 141)

In software, measurement is very important for managing software life cycle activities. ISO/IEC 15939 is an international standard that defines a measurement process, as well as the measurement information model and associated terminology (ISO/IEC, 2002). This standard can be used by a supplier or acquirer to implement a measurement process or to evaluate the conformance of a measurement process with respect to this standard (ISO/IEC, 2002, p. 1). The measurement model, according to ISO/IEC 15939, is divided into three parts, as shown in Figure 5: information need, measurable concept, and entity. Format definitions of these concepts can be found in 3.1.1.3. The model "helps to determine what the measurement planner needs to specify during measurement planning, performance, and evaluation" (ISO/IEC, 2002, p. 19).

The Practical Software Measurement (PSM) document can be used as an implementation guide for this standard issue (McGarry *et al.*, 2001).

1.10 What is performance measurement?

Performance measurement "is often discussed[,] but rarely defined. Literally[,] it is the process of quantifying action, where measurement is the process of quantification and action leads to performance" (Neely *et al.*, 1996).

According to Neely, the references to performance measurement most often cited come from the fields of accounting, information systems, operations management, and operations research (Neely, 2005).
In (GAO, 2005), the U.S. General Accounting Office (GAO) provides the following definition: "Performance measurement is the ongoing monitoring and reporting of program accomplishments, particularly progress toward preestablished goals."

The role of performance measurement is to "motivate behavior, leading to continuous improvement in customer satisfaction, flexibility, and productivity" (Lynch and Cross, 1995, p. 1).

1.11 What is performance management?

A performance management approach allows an organization to be more efficient and more effective by establishing organizational targets and continuously analyzing those targets to adapt them to changes.

Performance management is defined by the United States Office of Personnel Management as “the systematic process by which an agency involves its employees, as individuals and members of a group, in improving organizational effectiveness in the accomplishment of agency mission and goals” (Management, 2008).

According to Sink and Tuttle, managing organizational performance means to create visions in order to know ‘what the desired future state is’ (Sink and Tuttle, 1989, p. 34):

> ”Planning – assessing present organizational status relative to the vision, creating strategies for how the desired future state can be attained, and building on strengths so as to move toward the vision; Designing, developing, and effectively implementing specific improvement interventions that have a high probability of moving us toward the desired future state, particularly in terms of levels of performance; Designing, redesigning, developing, and implementing measurement and evaluation systems […] Ensuring that cultural support systems are in place […]” (Sink and Tuttle, 1989, p. 34).

According to Bourne, Franco, and Wilkes, performance management is a term associated with organizations reviewing and managing individuals, and they argue that there is:

> “growing concern in performance measurement that measuring performance is not enough. Measurement has to lead to insight and insight to action – hence the term corporate performance management has been born to differentiate between management at the level of the individual and the corporation” (Bourne *et al.*, 2003).

According to Flamholtz and Randle, an effective PMF should include (Flamholtz and Randle, 1998, pp. 233-238):

- a definition of objectives (what is expected) and goals (the level of performance expected);
- measurements (what is measured and how it is measured);
- an analysis of progress by review and feedback (analyze the goals and make adjustments to achieve them);
- a performance appraisal (an evaluation of performance);

- rewards (financial or other incentives for performance).

1.12 What are the key measures of performance management?

According to Artley and Stroh, performance measures can be looked at from six viewpoints: effectiveness, efficiency, quality, timeliness, productivity, and safety (Artley and Stroh, 2001).

According to (Sink, 1985; Sink and Tuttle, 1989), there are at least seven distinct, although not necessarily mutually exclusive, measures of "organizational system" performance. These are: effectiveness, efficiency, quality, profitability, productivity, quality of work life, and innovation.

Effectiveness: Are we doing the right things? The operational definition of effectiveness, as shown in Figure 1.4, is the "accomplishment of the 'right' things, and the value of this is measured by dividing actual output (AO) by expected output (EO). Most frequently, two attributes further define effectiveness: timeliness and quality" (Sink and Tuttle, 1989, p. 171). An upstream system, as presented in Figure 1.4, consists of the suppliers, vendors, customers, and procurement processes that provide input to the transformation process. This process is the object for which we are measuring performance, and downstream systems are the customers and markets that we wish to serve, or are obliged to serve.

According to Sink, at least three criteria should be used to analyze effectiveness (Sink, 1985, p. 42): quality (do the 'right' things to conform to the requirements), quantity (get all the 'right' things done), and timeliness (get the 'right' things done on time).

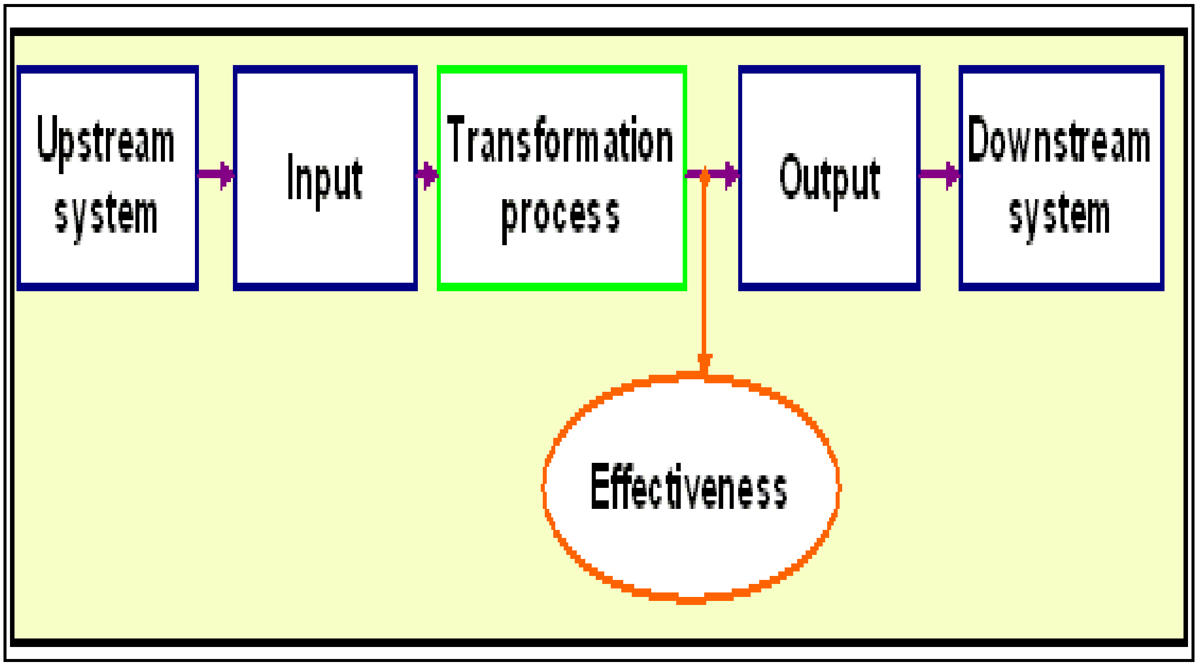


Figure 1.4 Effectiveness
Adapted from Sink and Tuttle (1989, p.171)
(Reprinted with the permission of the Institute of Industrial Engineers, 3577 Parkway Lane, Suite 200, Norcross, GA 30092, www.iienet.org. Copyright©2010 by the Institute of Industrial Engineers)

Artley and Stroh explain effectiveness as "a process characteristic indicating the degree to which the process output (work product) conforms to requirements" (Artley and Stroh, 2001, p. 4 Volume 2).

Efficiency: Are we doing things right? Efficiency is linked to input variables. Sink and Tuttle define it, as shown in Figure 1.5, as "resources expected or predicted or forecasted or estimated to be consumed divided by resources actually consumed." (Sink and Tuttle, 1989, p. 172).

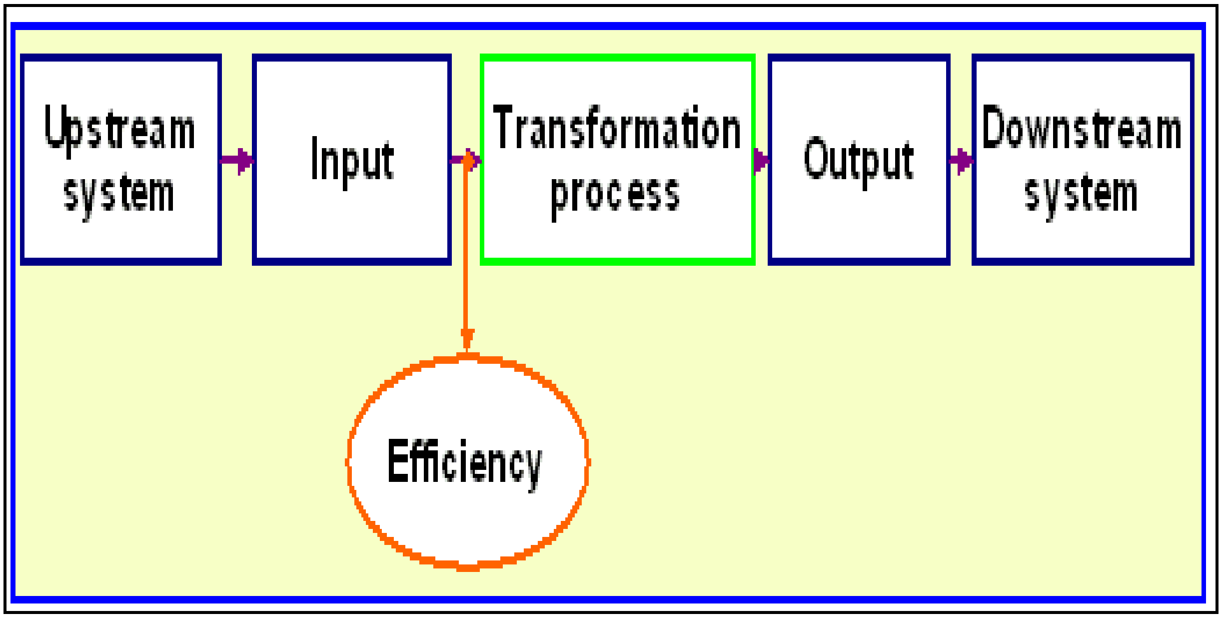


Figure 1.5 Efficiency.
Adapted from Sink and Tuttle (1989, p.172)

Artley and Stroh define efficiency as “a process characteristic indicating the degree to which the process produces the required output at minimum resource cost” (Artley and Stroh, 2001, p. 4 Vol. 2).

Profitability: Sink and Tuttle define profitability, as shown Figure 1.6, as “a relationship between total revenues and total costs” (Sink and Tuttle, 1989, p. 185).

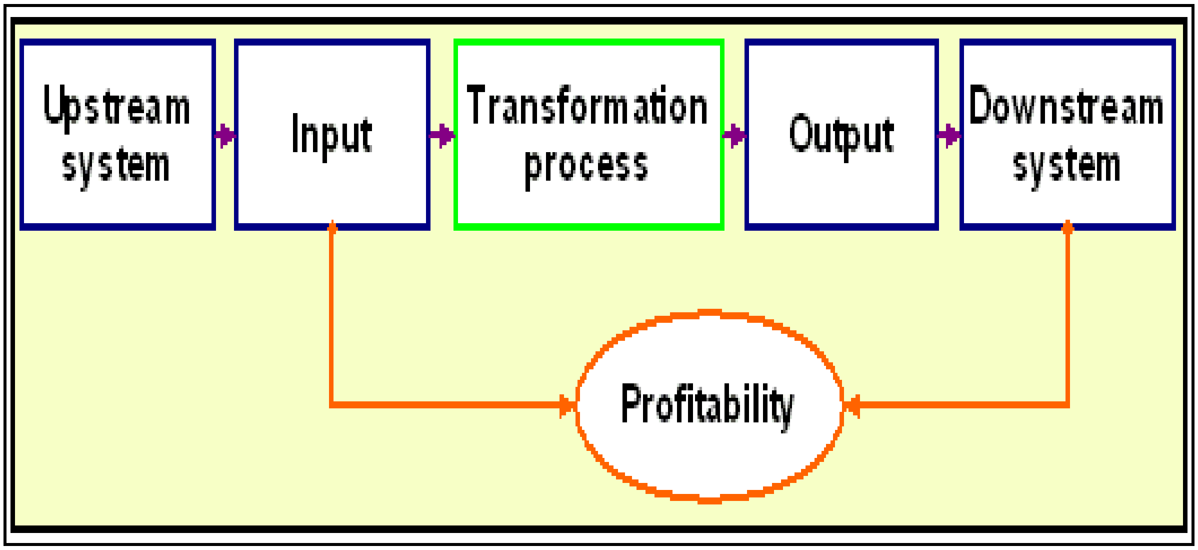


Figure 1.6 Profitability
Adapted from Sink and Tuttle (1989, p.185)
(Reprinted with the permission of the Institute of Industrial Engineers, 3577 Parkway Lane, Suite 200, Norcross, GA 30092, www.iienet.org. Copyright©2010 by the Institute of Industrial Engineers)

According to Sveiby profit is not a good yardstick to compare organizations with large intangible assets. 'Profit margin' is described as an important indicator in a knowledge organization. Profit as a percentage of sales, profit as a percent of added value, etc. would be better indicators (Sveiby, 1997b, pp. 153-154)

Quality is defined by Sink and Tuttle as "the degree to which a product or service meets customer requirements and expectations." (Sink and Tuttle, 1989, p. 172).

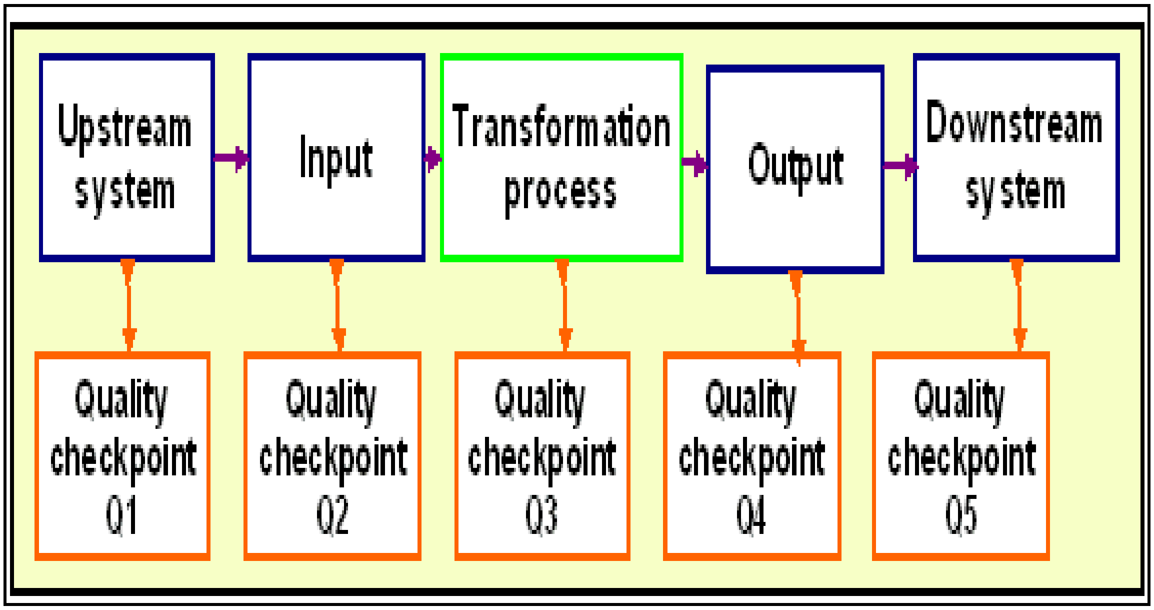


Figure 1.7 Quality.
Adapted from Sink and Tuttle (1989, p.172)
(Reprinted with the permission of the Institute of Industrial Engineers, 3577 Parkway Lane, Suite 200, Norcross, GA 30092, www.iienet.org. Copyright©2010 by the Institute of Industrial Engineers)

The same definition is given by Artley and Stroh (Artley and Stroh, 2001, p. 4 Vol. 2). According to Crosby, the main problem with quality management is that the manager "does not provide a clear definition of quality, so the employees each develop their own." (Crosby, 1995, p. 3). It is necessary to evaluate software quality, because "the software product may be hard to understand and difficult to modify…difficult to use or easy to misuse…may be unnecessarily machine-dependent, or hard to integrate with other programs" (Boehm *et al.*, 1976). Non-functional quality factors could be critical, and "one of the most critical non-functional quality factors is the performance characteristic of a software system" (Schmietendorf *et al.*, 2000).

Software quality is a constant topic of concern in the software industry. It is an elusive characteristic not only because it is difficult to achieve, but also because it is difficult to describe. There are a variety of ways of viewing or defining quality, as illustrated by the definitions above, and the SWEBOK Guide describes a number of "ways of achieving software quality" (Abran *et al.*, 2004, p. 11) .

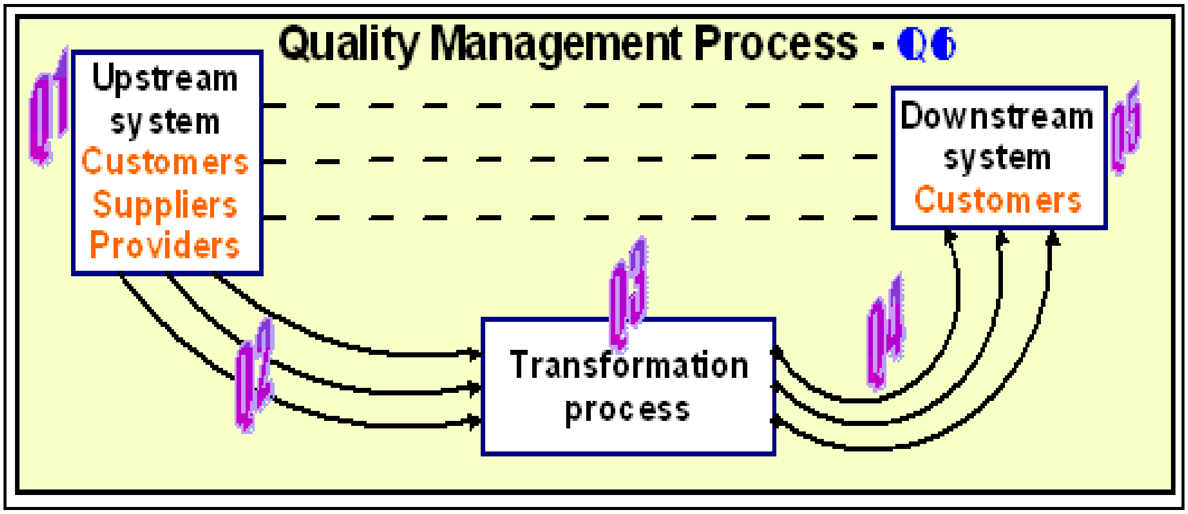


Figure 1.8 Six quality checkpoints.
Adapted from Sink and Tuttle (1989, p.179)
(Reprinted with the permission of the Institute of Industrial Engineers, 3577 Parkway Lane, Suite 200, Norcross, GA 30092, www.iienet.org. Copyright©2010 by the Institute of Industrial Engineers)

Quality has to be operationally defined and measured for at least five quality checkpoints (Q1 to Q5), as shown in Figure 1.7 (Sink and Tuttle, 1989, p. 172). Q6 is a quality checkpoint with respect to the coordination and overall management of the quality management process, as shown in Figure 1.8 (Sink and Tuttle, 1989, p. 179). Quality is a pervasive aspect of the performance of an organization that must be measured and managed at all checkpoints.

In the context of software engineering (IEEE Computer Society, 1990), quality is defined as "the degree to which a system, component, or process meets specified requirements" or "the degree to which a system, component, or process meets customer or user needs or expectations."

Productivity is output over input, as shown in Figure 1.9. Sink and Tuttle believe that "productivity is an important criterion of performance because, when you measure it well, you end up learning something about effectiveness, efficiency, and quality" (Sink and Tuttle, 1989, p. 180).

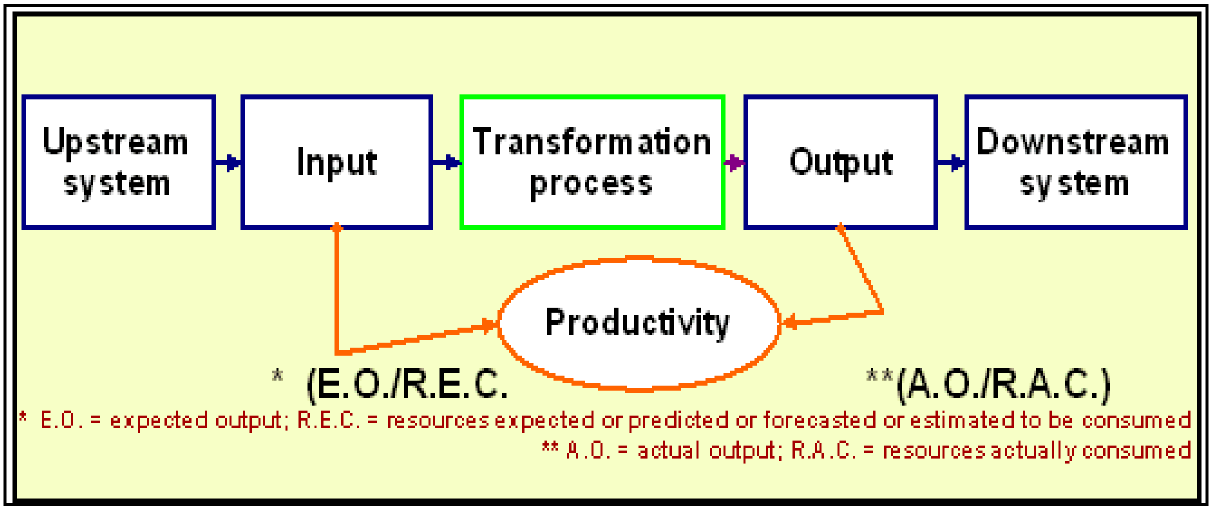


Figure 1.9 Productivity.
Adapted from Sink and Tuttle (1989, p.180)
(Reprinted with the permission of the Institute of Industrial Engineers, 3577 Parkway Lane, Suite 200, Norcross, GA 30092, www.iienet.org. Copyright©2010 by the Institute of Industrial Engineers)

According to Artley and Stroh, productivity is defined as "the value added by the process divided by the value of the labor and capital consumed" (Artley and Stroh, 2001, p. 4 Vol. 2).

In software development, productivity is one of the most studied aspects (Anselmo and Ledgard, 2003), and is expressed as the ratio between the size of the software and the effort necessary to realize it. Effort is the total effort expended to accomplish the task (usually in person-hours, -days, -months, or -years).

Function Point Analysis (FPA), promoted by the International Function Point Users Group (IFPUG), is one of the most widely used functional size measurement methods for software (IFPUG, 2008; ISO/IEC, 2003b). The Common Software Measurement International Consortium (COSMIC) (COSMIC, 2008) proposed an improved functional size measurement method for software, initially known as Full Function Points and has now evolved into COSMIC (Abran *et al.*, 1998; COSMIC, 2008; ISO/IEC, 2003a). Functional size measurement (FSM) is key to the ISBSG repository, which is discussed in section 2.4.

Quality of Work Life describes how people feel about various aspects of their work life and is critical to organizational performance. It is related to the Transformation process, as shown in Figure 1.10.

It is defined as "the affective response or reaction of the people…in planning, problem solving, and decision making" (Sink and Tuttle, 1989, p. 182).

According to Artley and Stroh, Quality of Work Life is called 'safety', and is defined as "the overall health of the organization and the working environment of its employees" (Artley and Stroh, 2001, p. 4 Vol. 2).

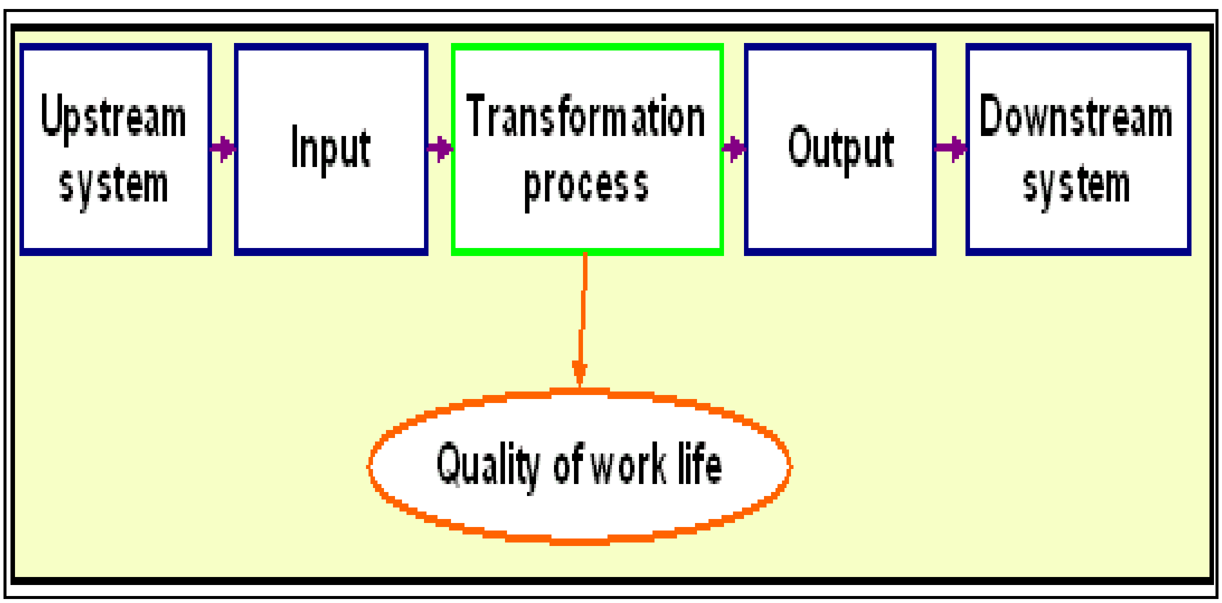


Figure 1.10 Quality of work life.
Adapted from Sink and Tuttle (1989, p.182)

Innovation also concerns the transformation process, as shown in Figure 1.11. This can be described as "the creative process of changing what we are doing, how we are doing things, technology, products, services, methods, procedures, policies…to successfully respond to internal and external pressures, opportunities…" (Sink and Tuttle, 1989, p. 183). Through the process of innovation, there will be new, better, and more functional products and services: without innovation, it is difficult to compete.

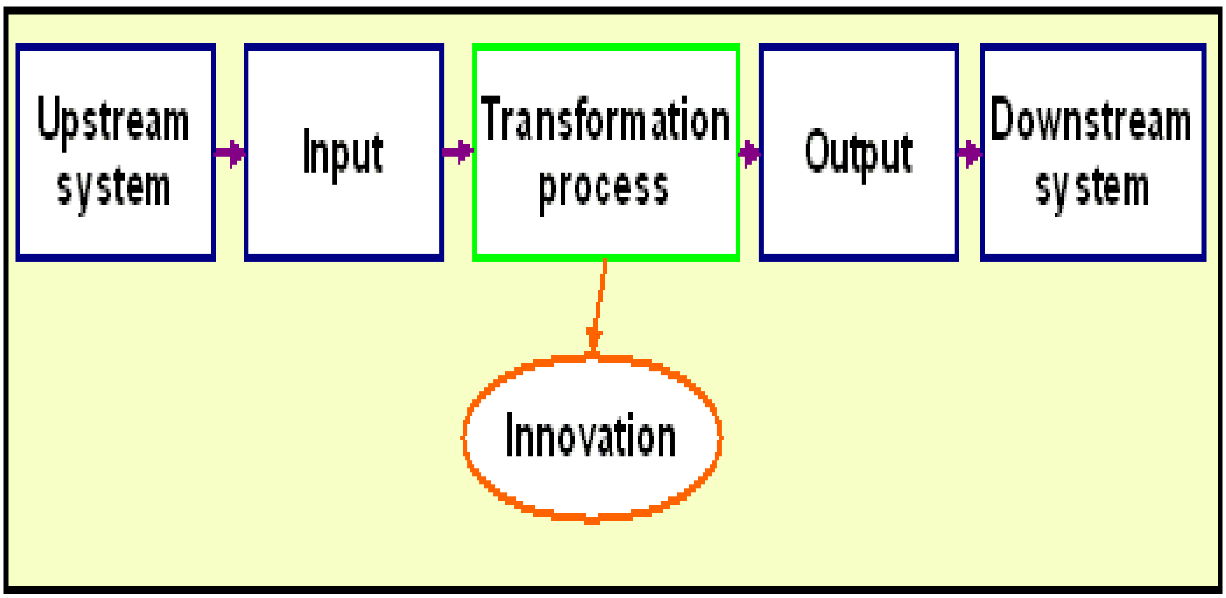


Figure 1.11 Innovation.
Adapted from Sink and Tuttle (1989, p.183)

1.13 Summary

This chapter introduced basic concepts and terms that are essential to understanding this thesis and shows how they are viewed in it as well. These concepts and terms, which are included in the title of the thesis, have been discussed in thirteen subsections of this chapter, as shown in Figure 1.2.

CHAPTER 2

PERFORMANCE MANAGEMENT FRAMEWORKS AND THE ISBSG REPOSITORY

In the previous chapter, the key concepts of this thesis were defined and explained. In this chapter, a synthesis of the literature related to performance management is presented. This is followed by an analysis of the QEST prototype and a review of the ISBSG repository, as shown in Figure 2.1.

The selected subset of multidimensional performance management models or frameworks presented in this chapter was chosen because these models were deemed to be representative of the wider set of models found in the literature and because of the possibility of applying elements of those models in the PMF discussed in chapter 3 and in the realization of the prototype presented in chapter 4. Please note that the terms 'framework' and 'model' are used interchangeably.

First, section 2.1 presents a synthesis of the literature related to multidimensional performance models in software engineering management. Then, section 2.2 presents a synthesis of the literature related to multidimensional performance models in management in general. The well-known BSC is presented, as are other known models in management, such as EFQM, Baldrige, IAM, PRISM, and the PMF developed by Sink and Tuttle, one of the first available in the literature. Section 2.3 presents a comparison of these models, through a summary table and its accompanying discussion. Section 2.4 presents and analyzes the repository of the ISBSG. Section 2.5 assesses the strengths and weaknesses of the previously developed QEST prototype. The final section, section 2.6, presents a discussion of the previous sections and an analysis of the strengths and weaknesses of the models presented in this chapter.

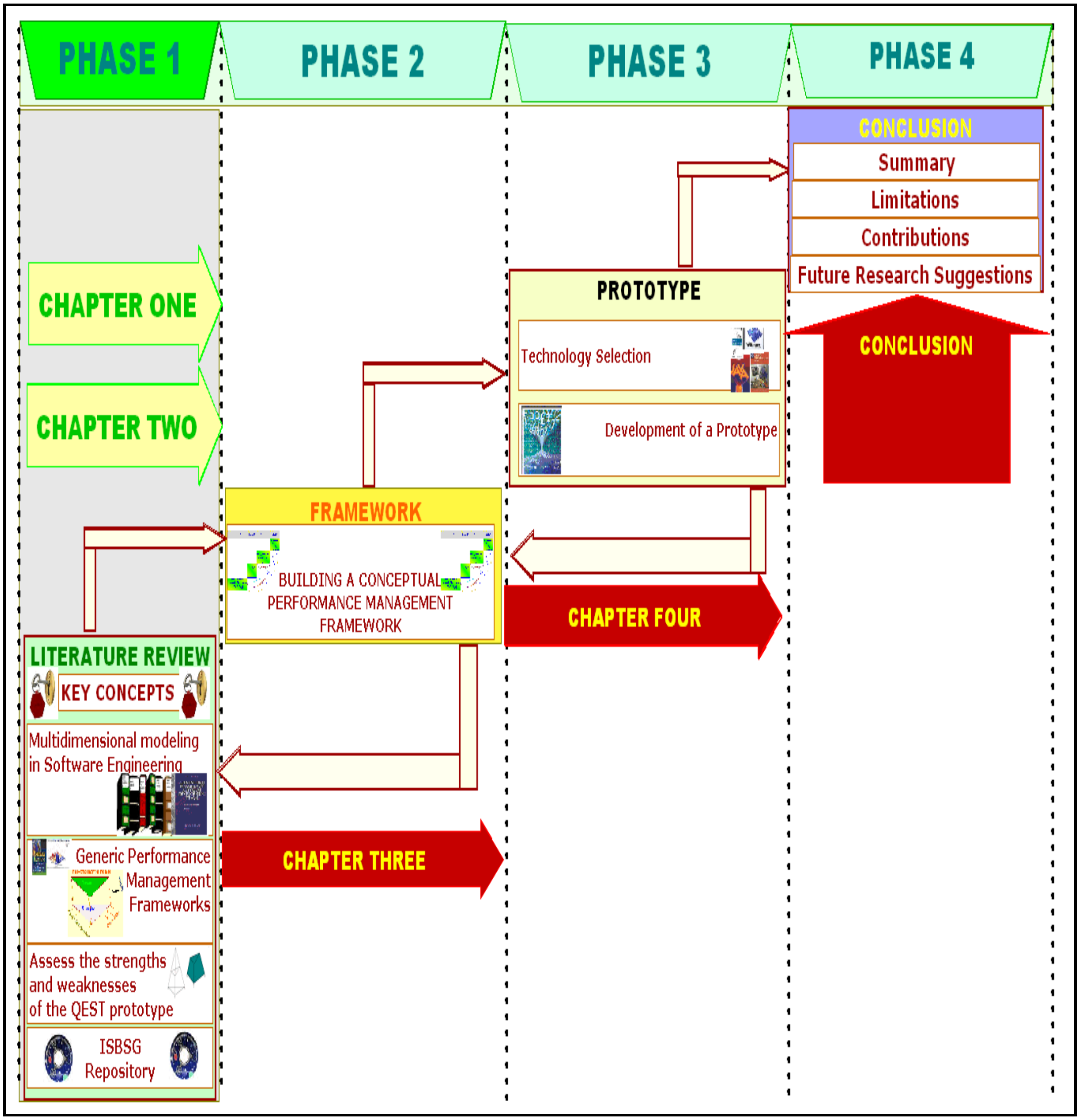


Figure 2.1 Thesis – literature review – Phase 1.

2.1 Performance Management Frameworks in software engineering

This section describes PMFs or models in software engineering. These models concern performance or quality. The models are all presented using the same template that is described in Table 2.1.

Table 2.1 Template for identifying and analyzing performance models

Identify	Model name	• The name of the model
	Reference	• The published work that was identified as the source of the model • The bibliometric method of evaluation and its associated value for this publication (this is one approach to identifying the influence of this model in the literature). The citation index will be used for the most relevant references only.
	Origin	• The origin of the model: authors and date
	Purpose	• What is the purpose of the model?
	Usage	• Who is using the model?
Analyze	Viewpoint Indicators	• How many viewpoints does the model support? • What measures, indicators, and viewpoints are included in the model? • What kind of indicators (qualitative and/or quantitative, economic, and/or non-economic) are used? How do they interrelate to obtain a single performance value?
	Visualization	• How are the indicators and viewpoints represented graphically? • Is there any prototype/tool that supports the model?
	Note	• What other element is of interest?

2.1.1 McCall model

The McCall model shown in Figure 2.2 and Table 2.3, also called the GE model or FCM (Factor-Criteria-Metric), was proposed in 1978 (Cavano and McCall, 1978).

It groups software quality characteristics according to the main phases in the software life cycle. These high-level categories of quality characteristics are:

- product operation: quality characteristics related to using the product;
- product revision: quality characteristics related to maintaining the product;
- product transition: operational characteristics (quality characteristics related to porting the product to a new environment).

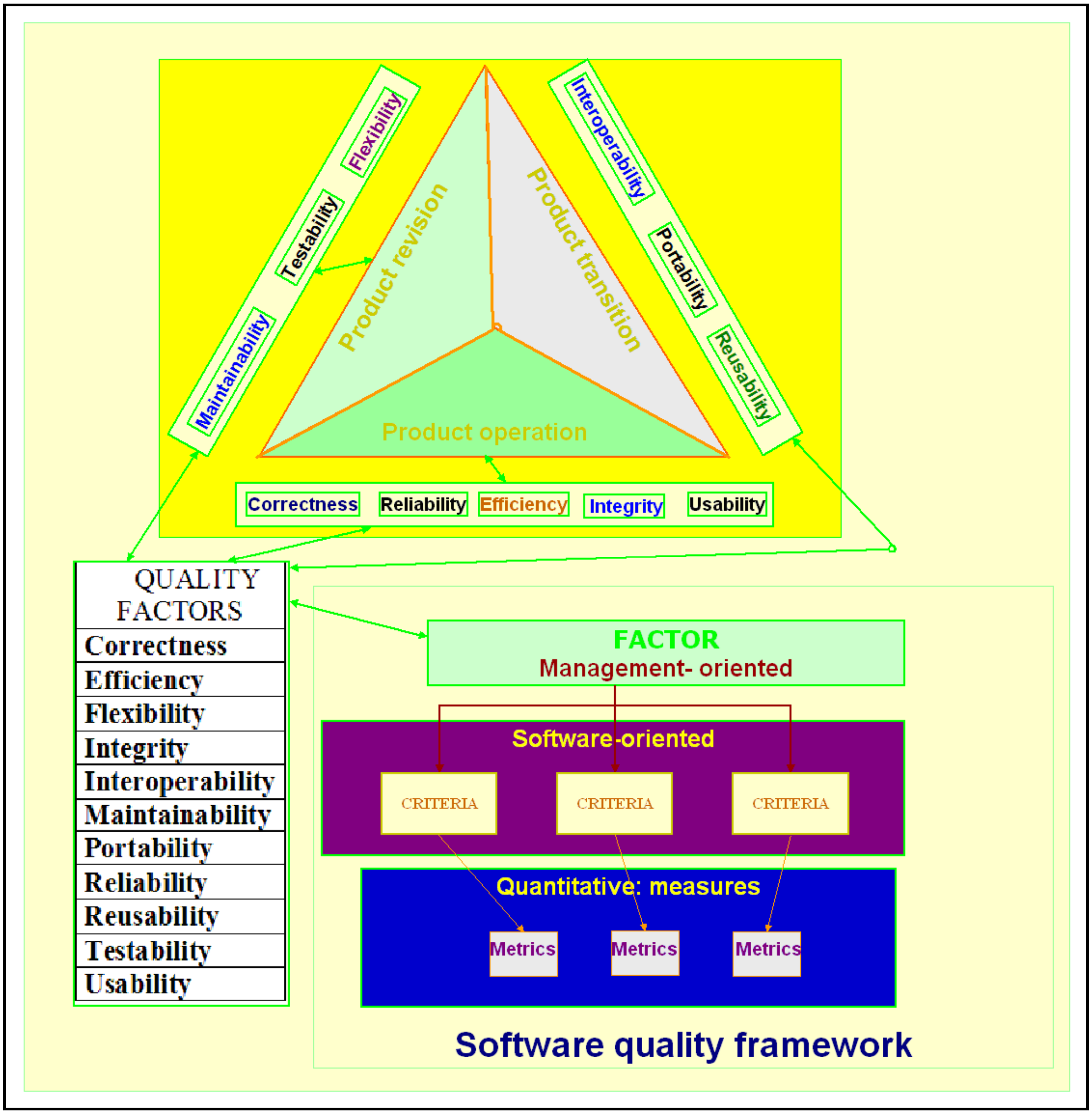


Figure 2.2 McCall model.
A synthesis of Alonso et al.(1998; p.4855) and Cavano and McCall(1978, p. 135)

Factors (high-level quality characteristics) that are difficult to measure are broken down into lower-level quality indicators (Criteria) that are more tangible and measurable, as shown in Figure 2.2, where three different levels are identified:

- Factors: quality characteristics from the viewpoint of users – the objectives of users;
- Criteria: quality characteristics from the viewpoint of developers – the objectives of developers;
- Metrics: the lowest level – where quality characteristics will be measured.

For each criterion, it suggests one or more measurements (metrics) for measuring the degree to which the system possesses or exhibits the associated quality criteria. This model defines various quality factors (Correctness, Efficiency, Flexibility, Integrity, Interoperability, Maintainability, Portability, Reliability, Reusability, Testability, and Usability) that describe the external views of the system, as they are perceived by end-users. As shown in Table 2.2, each factor is then decomposed into criteria that describe the internal views of the product as they are perceived by software developers. A given criterion can be common to several factors. One of the major contributions of the McCall model is the relationship created between factors and criteria.

Alonso and others (Alonso *et al.*, 1998) have presented a new quality model, the purpose of which is to evaluate the quality of software built using object-oriented techniques. The model is based on the framework defined by McCall. The criteria are analyzed one by one to find a relationship between the criterion and a quality attribute of an object-oriented system. They have found that all criteria represent a characteristic that object-oriented systems should possess.

However, Alonso also claims that the 23 criteria determined by McCall are not enough to cover the features of an object-oriented system, and, after examining the literature, they added three more criteria: documentation, stability, and structuredness, as shown in Table 2.2 (Alonso *et al.*, 1998).

Table 2.2 Relationship of criteria to software quality factors
Adapted from Alonso (1998, p. 4885)

QUALITY FACTORS	QUALITY CRITERIA
Correctness	Completeness, Consistency, Traceability
Efficiency	Execution Efficiency, Storage Efficiency
Flexibility	Documentation, Expandability, General Modularity, Self-Descriptiveness, Simplicity, Stability, Structuredness
Integrity	Access Audit, Access Control
Interoperability	Documentation, Communications Commonality, Data Commonality, Modularity, Structuredness, Traceability
Maintainability	Documentation, Conciseness, Consistency Modularity, Self-Descriptiveness, Software System Independence, Structuredness
Portability	Documentation, Machine Independence, Modularity, Self-Descriptiveness, Software System Independence, Structuredness
Reliability	Accuracy, Completeness, Consistency, Error Tolerance, Simplicity
Reusability	Documentation, Generality, Modularity, Self-Descriptiveness, Software System Independence, Structuredness
Testability	Communicativeness, Documentation, Instrumentation, Modularity, Self-Descriptiveness, Simplicity, Structuredness
Usability	Documentation, Communicativeness, Operability, Training

Table 2.3 Summary – McCall model

Model Name	The McCall model (also known as the General Electric Model)
Reference	(Cavano and McCall, 1978): ACM Bibliometrics – Citation Count: - 8 (15.08.2010) (Alonso *et al.*, 1998) (Fitzpatrick, 1996) Google : Citation Count- 10 (16.11.2010)
Origin	Developed by Jim McCall and updated by Cavano (Cavano and McCall, 1978)
Purpose	The purpose of the McCall model is to manage software quality.
Usage	"McCall's model is used in the United States for very large projects in the military, space and public domain. It was developed in 1976-1977 by the US Airforce Electronic System Division (ESD), the Rome Air Development Centre (RADC) and General Electric (GE)" (Fitzpatrick, 1996).
Viewpoints	Number of viewpoints: 3
	• **product revision**: ability to undergo changes, linked to error correction and system adaptation • **product transition**: adaptability to new environments, linked to error correction and adaptation to a new environment • **product operations**: its operational characteristics, linked to the capacity to provide the results required by the user, to be operated efficiently, and.

Table 2.3 Summary – McCall model (continued)

Viewpoints	to be quickly understood
Indicators	• 11 Factors (Correctness, Efficiency, Flexibility, Integrity, Interoperability, • Maintainability, Portability, Reliability, Reusability, Testability, and Usability): the external view as seen by the users • 23 quality criteria (measures): the internal view, as seen by the developers • 3 more quality criteria (measures) added by Alonso: documentation, stability, and structuredness, as shown in Table 2.2
Visualization	▪ No visualization approaches or techniques are suggested.
Notes	• One of the best known quality models ▪ It is odd that this quality model does not seem to include any factor or criterion directly related to the actual functionality delivered by a given piece of software. ▪ It does not include a single or unifying measurement index to quantify quality.

2.1.2 Boehm model

As is the case with the McCall model, the Boehm model is one of the first models of software quality ever proposed (Boehm, 1978). It proposes a multilevel hierarchy, as shown in Figure 2.3. Boehm claims that a software product is usable if it is portable and maintainable, and enables 'as is' utility.

As shown in Table 2.4, the Boehm model is similar to the McCall model, in that it has adopted a hierarchical structure. Boehm identifies a hierarchy with three high-level characteristics linked to 7 factors that are themselves linked to 15 measures.

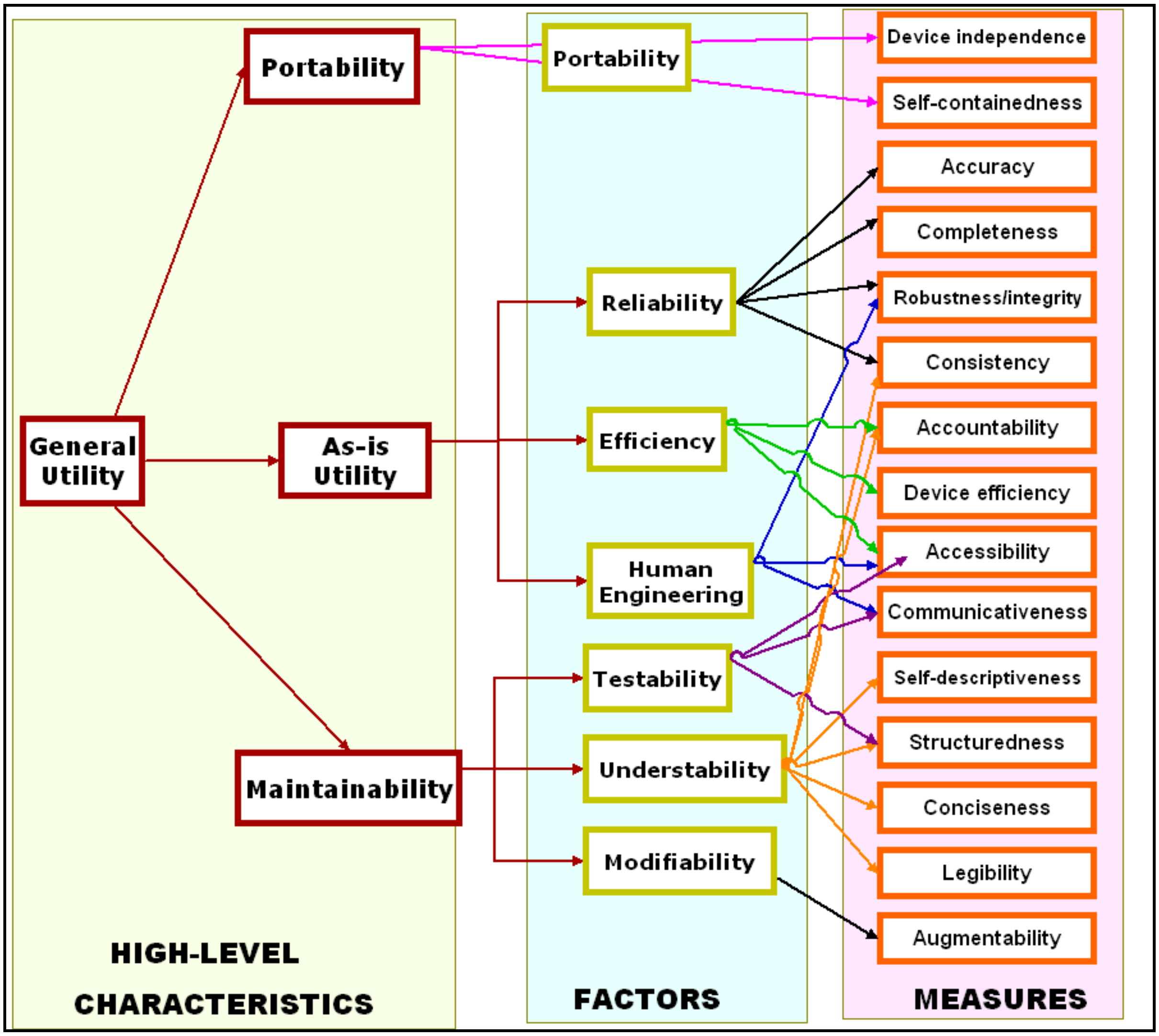


Figure 2.3 The Boehm's quality model.
Adapted from Boehm (1978, p. 595)
(Copyright 2008 the University of Southern California)

Table 2.4 Summary – Boehm model

Model Name	The Boehm Model
Reference	(Boehm *et al.*, 1976) ACM Bibliometrics 26.01.2009: Citation Count – 40 and Google : Citation Count- 309 (16.11.2010) (Boehm, 1978) Amazon.com: 11 books cite this work (02.11.2010)
Origin	TRW Systems and Energy (TRW, 2009)
Purpose	The purpose of the Boehm model is to manage quality via a set of attributes and measurements.

Table 2.4 Summary – Boehm model (continued)

Usage	Management of the quality of software in a quantitative and qualitative manner using a given set of criteria and metrics
Viewpoints	No. of Viewpoints: 3
	Portability – Can I still use it if I change the configuration? Could it operate easily and well on configurations other than the original one? **As Is Utility** – How easily and effectively can it be used as is? This is defined by reliability, efficiency, and human engineering: • Reliability: performs its intended functions satisfactorily • Efficiency: "The code possesses the characteristic efficiency to the extent that it fulfills its purpose without waste of resources" (Boehm *et al.*, 1976). • Human Engineering: "The code possesses the characteristic usability to the extent that it is reliable, efficient, and human-engineered" (Boehm *et al.*, 1976). **Maintainability** – How easy is it to understand, test, and modify? This is defined by testability, understandability, and modifiability: • Testability: "The code possesses the characteristic testability to the extent that it facilitates the establishment of verification criteria and supports evaluation of its performance" (Boehm *et al.*, 1976). • Understandability: "The code possesses the characteristic understandability to the extent that its purpose is clear to the inspector" (Boehm *et al.*, 1976). • Modifiability: "The code possesses the characteristic modifiability to the extent that it facilitates the incorporation of changes, once the nature of the desired change has been determined" (Boehm *et al.*, 1976).
Indicators	7 Factors and 15 measures, as shown in Figure 2.3
Visualization	No visualization approaches or techniques are suggested.

2.1.3 ISO 9126 quality model

The International Organization for Standardization introduced a standard, Software Product Evaluation Quality Characteristics and Guidelines for Their Use, to interpret and measure software quality (ISO/IEC, 2001).

In 1987, only 8% of IT representatives knew about AQAP-13 (NATO Software Quality Control System Requirements), which was perhaps the most important software quality standard at that time, unlike the 70% of interviewees who had at least heard of ISO 9126 in 1993 (Bazzana *et al.*, 1993).

As shown in Table 2.5, the quality model defined in ISO/IEC 9126-1 (ISO/IEC, 2001) recognizes three viewpoints of software quality, and defines them as follows :

- **Quality in use** is the user's view of the software product in a specific context of use, as shown in Figure 2.4 (a low level of quality of use means that the system is not used at all or is used only partially)**.**
- **External quality** is the external view of the software product, as shown in Figure 2.5 . It is applicable to software that is running: the software product is evaluated during its execution by the user.
- **Internal quality** is the internal view for all the characteristics measured and evaluated against requirements, as shown in Figure 2.5 . It can be measured by the developer during the software life cycle.

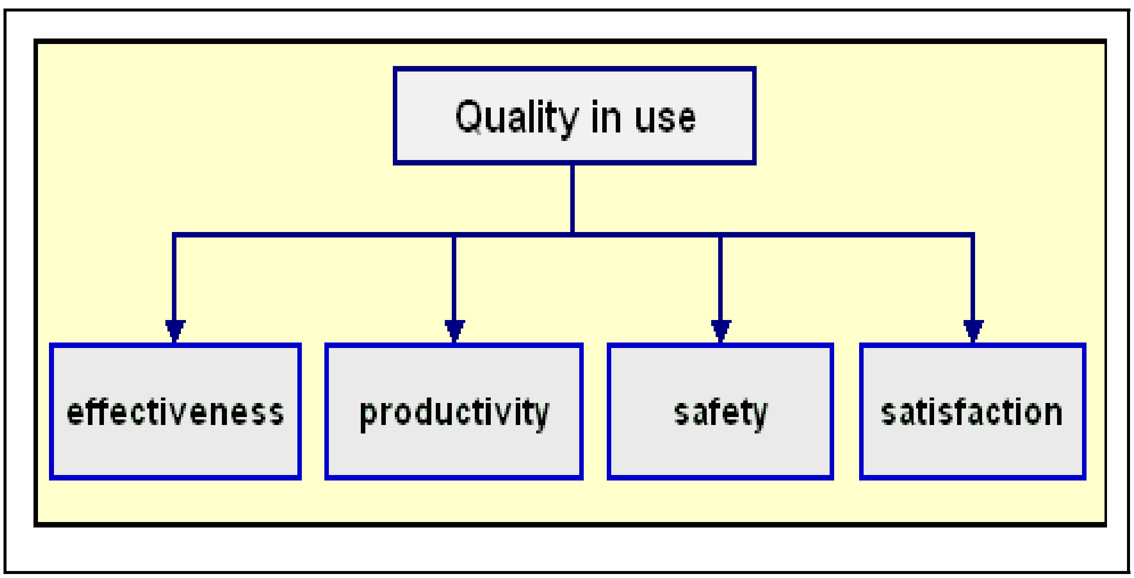


Figure 2.4 Quality in use model.
Adapted from ISO/IEC (2001, p. 12)

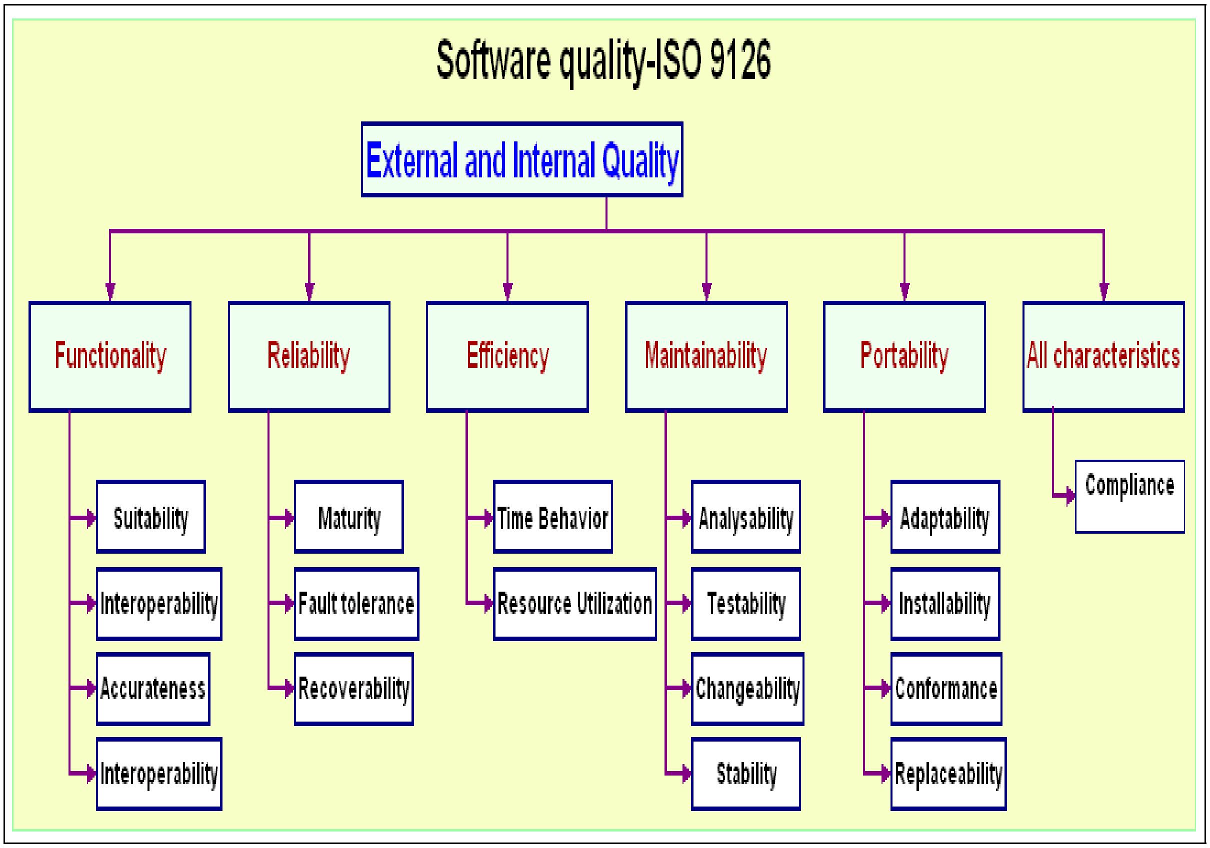


Figure 2.5 ISO 9126 – External and internal quality.
Adapted from ISO/IEC (2001, p.7)

Table 2.5 Summary – ISO 9126 model

Model Name	Quality model of the ISO 9126 Standard
Reference	(ISO/IEC, 2001) (Bazzana *et al.*, 1993)
Origin	International standard
Purpose	The purpose of the model is to interpret and measure software quality.
Summary of structure	The hierarchy is strict: each high level quality characteristic is related to exactly one set of sub-characteristics.
Usage	It is the software product evaluation standard of the International Organization for Standardization.
Viewpoints	No. of viewpoints: 3
	Quality is divided into three viewpoints: external quality, internal quality, and quality in use.

Table 2.5 Summary – ISO 9126 model (continued)

Indicators	▪ Internal and external quality have 6 indicators (characteristics: functionality, reliability, efficiency, maintainability, portability, and all characteristics) and 18 sub-characteristics, as shown in Figure 2.5. • Quality in use has 4 characteristics: effectiveness, productivity, safety, and satisfaction.
Visualization	No visualization approaches or techniques are suggested.
Notes	▪ It does not propose any technique for handling more than one viewpoint at a time.

2.1.4 Donaldson and Siegel model

According to Donaldson and Siegel (Donaldson and Siegel, 1998; 2001), *n* different normalized measures are used to define the 'product integrity index' as a dimension (viewpoint) using a vectorial approach, as shown in Figure 2.6, Figure 2.7, and in Table 2.6. Product integrity is defined in terms of product attributes and attribute value scales. The authors claim that this model enables measurement of the "goodness" of the products and the "goodness" of the software system development process.

The concept of length is used to define the 'product integrity value'. The Product Integrity (PI) vector is the distance between the two green points, as shown in Figure 2.7. The starting point is situated at the origin of the coordinate system and the head of the blue line represents the end point, as shown in Figure 2.7.

The PI vector for n viewpoints PI=$\sqrt{\sum_{i=1}^{n} L_i^2}$.

The PI index value is defined as follows:

$$\text{Equation 1: PIindex} = \frac{\sqrt{\sum_{i=1}^{n} L_i^2}}{\sqrt{\sum_{i=1}^{n} w_i^2 (\max([at_i]))^2}};$$

L = the length of the vector;

at = Product Integrity attribute;

$\max([at_i])$ = maximum value for at_i (value normalized from zero to one)

n = number of Product Integrity attributes;

w = weighting factor for at attributes (in the case of an unweighted average w=1, as shown in Figure 2.6 using a Kiviat graph)

'PIindex' is a multidimensional concept that uses a vector to represent the goodness of the products and the development process. A PIindex close to one means that the goodness is at its best, and close to zero means that it is at its worst.

The steps used to calculate PIindex are as follows:

- select the questions that need to be addressed;
- select products to measure;
- identify product attributes;
- define a value scale for each attribute (use terms that are familiar to the organization);
- calculate the value of the product integrity index using the formula shown in Equation 1.

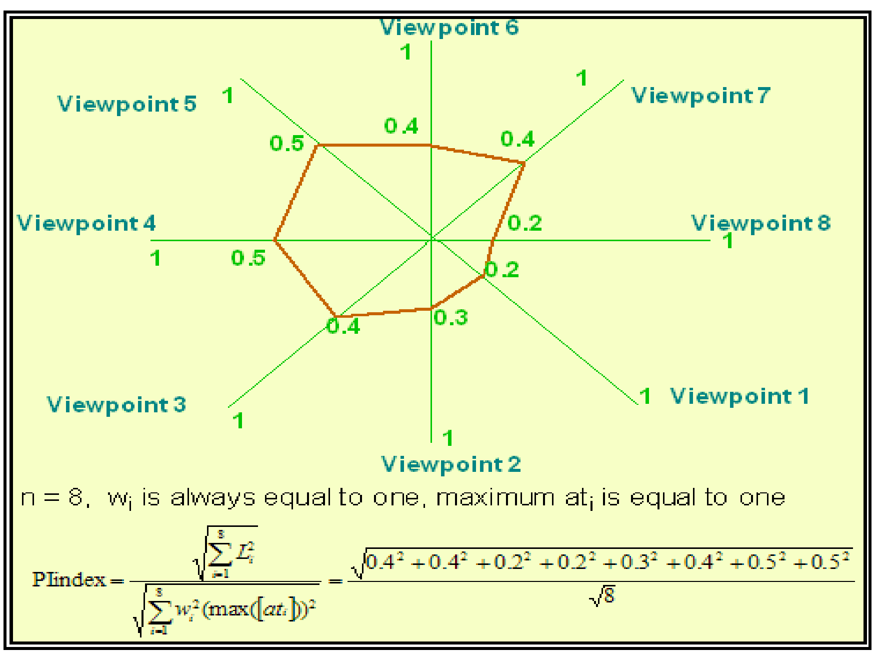


Figure 2.6 Visualization – the Kiviat graph with 8 dimensions (viewpoints).
A synthesis of (Donaldson and Siegel, 2001, pp. 381-446)

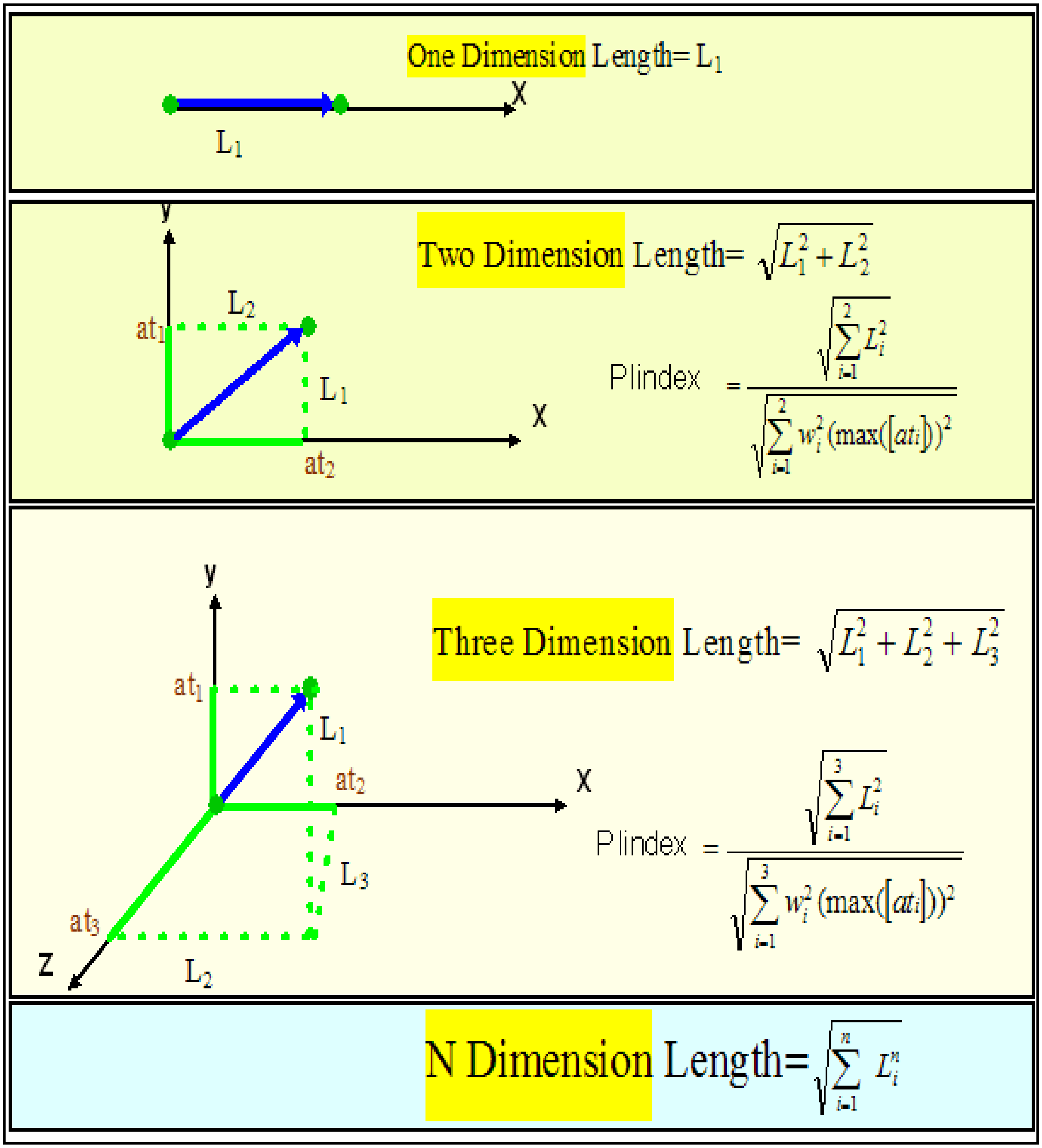


Figure 2.7 The vector length concept for combining multiple dimensions.
A synthesis of (Donaldson and Siegel, 2001, pp. 381-446)

Table 2.6 Summary – Donaldson and Siegel model

Model Name	The Donaldson and Siegel model
Reference	(Donaldson and Siegel, 1998; 2001) Amazon.com: 7 books cite this work (24.05.2010)
Origin	Science Applications International Corporation (SAIC): a large research and engineering company in the U.S. (Donaldson and Siegel, 2001).
Purpose	The purpose of the model is to measure the "goodness" of products and the "goodness" of the software system development process producing these products (Donaldson and Siegel, 2001).
Usage	SAIC: a Fortune 500 company, with approximately 45,000 employees, offices in over 150 cities worldwide, and annual revenues exceeding $5 billion (Donaldson and Siegel, 2001; SAIC, 2009)
Viewpoints	No. of viewpoints: *n* – generically defined
Indicators	*n*: generic and normalized measures
Visualization	The suggested visualization technique is a Kiviat graph, as shown in Figure 2.6.

2.1.5 Dromey model

Dromey proposes a working framework for building and using a quality model to evaluate requirements, design, and implementation artifacts (Dromey, 1996). It is a general model that relates the components of a software product to the high-level attributes that determine its quality, as shown in Table 2.7 and Figure 2.8. Two attributes, process-mature and reusability, are added to the ISO 9126 attributes (characteristics), as shown in Figure 2.5.

The fundamental axiom on which this approach is built states: "A product's tangible internal characteristics or properties determine its external quality attributes" (Dromey, 1996).

As shown in Figure 2.8, five steps are described to identify product properties, to link them to quality attributes, and to evaluate them.

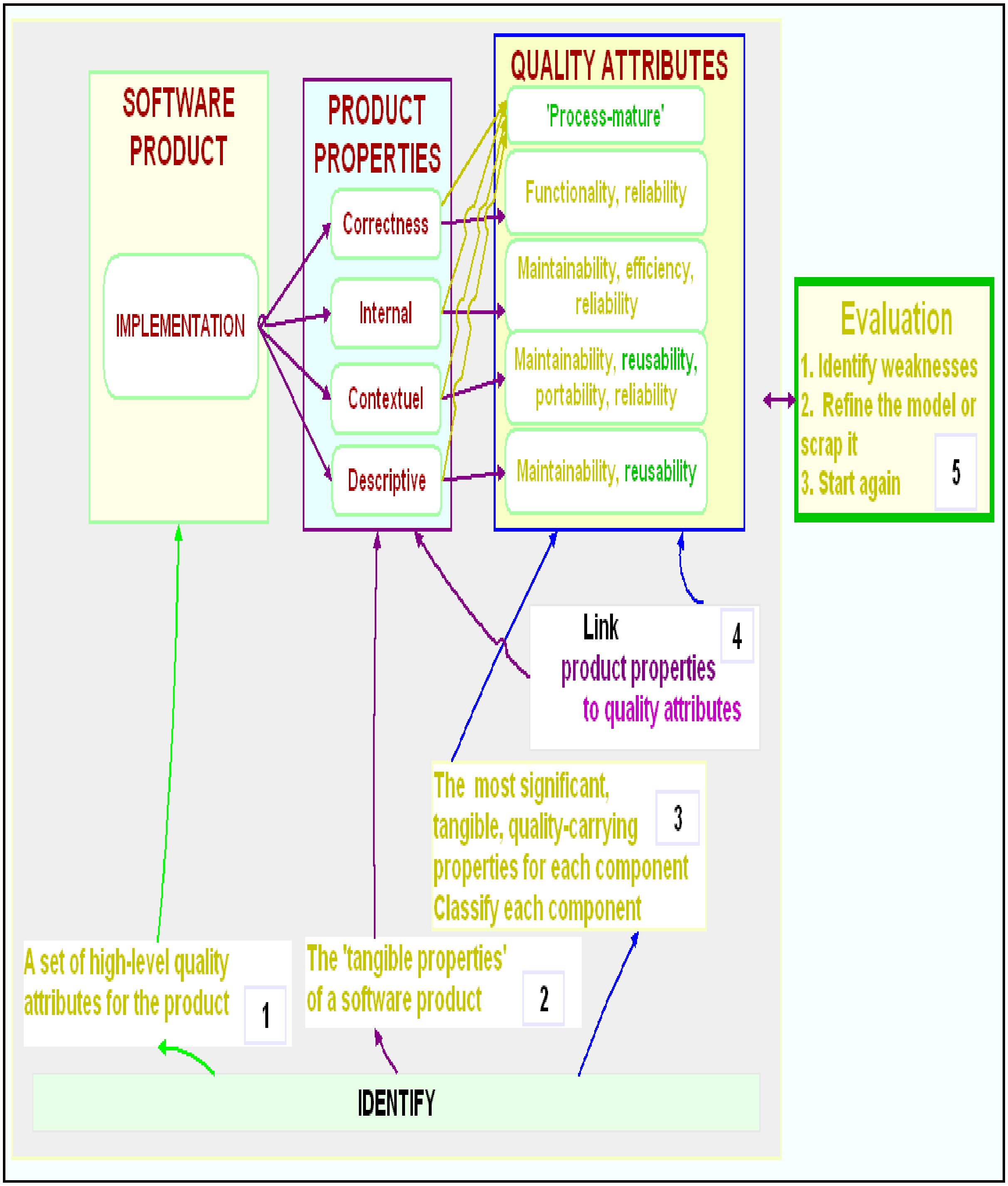


Figure 2.8 Linking product properties to quality attributes.
A synthesis of Dromey (1996, p.34-40)

Table 2.7 Summary – Dromey model

Model Name	The Dromey Model
Reference	(Dromey, 1996): ACM Citation Count – 21 (26.05.2010) and Google : Citation Count- 137 (14.11.2010) (Dromey, 1995): ACM Citation Count – 32 (26.05.2010) and Google : Citation Count – 209 (14.11.2010)
Origin	Published by Dromey in 1995-1996
Purpose	Generic software quality model
Usage	Assess and ensure software quality by linking tangible product properties to high-level quality attributes, as shown in Figure 2.8.
Viewpoints	No. of viewpoints: 4
	The model is based on four "tangible properties" of a software product: • **Correctness**: associated with individual components (internal) or with the way components are used in a given context (contextual). It is important to take into consideration that some properties are significant: "If they are violated the product will not perform as intended" (Dromey, 1996, p. 35). • **Internal**: measures how well a component has been deployed according to its intended use or implementation requirements. "Every component has a norma1 form that defines its internal 'truth'" (Dromey, 1996, p. 35) • **Contextual**: expresses a relation between a component and an external property of a component characterizing relational quality. • **Descriptive**: applies to requirements, designs, implementations, and user interfaces. "To be useful, a software product must be easy to understand and use for its intended purpose" (Dromey, 1996, p. 35)
Indicators	The main indicators are: **Functionality**: suitability, accuracy, interoperability, compliance and security; **Reliability**: maturity, fault-tolerance, recoverability, maintainability, reusability, portability; **Efficiency**: time behavior, resource behavior; **Usability**: understandability, learnability, operability **Maintainability**: analyzability, changeability, stability, testability; **Portability**: adaptability, installability, conformance, replaceability; **Reusability**: machine-independent, separable, configurable **Process-mature**: client-oriented, well-defined, assured, effective.
Visualization	No visualization approaches or techniques are suggested.

2.1.6 An Integrative Framework for IS quality management

According to Stylianou and Kumar (Stylianou and Kumar 2000), it is difficult to manage quality, because every product and service has many stakeholders with varying perceptions of quality.

This framework regards quality as multidimensional and as judged differently by diverse groups of stakeholders. It is addressed from the viewpoint of an IS manager and considers issues relating to multiple stakeholder groups, products, services, and process qualities. According to the authors, there is a “synergistic relationship” (Stylianou and Kumar 2000) between IS and non-IS business processes, and a clear link between measures to strategic goals.

Important issues to address in implementing IS quality are: customer focus (keeping customers satisfied with quality solutions and attending to their needs), a process approach (systematic management of resources, activities, and outcomes), leadership (be an example and display visionary leadership), culture, broad participation and teamwork, motivation, resources, training, measurement and constructive feedback, accountability for results, and rewarding achievements.
IS should identify all the quality concerns from various stakeholders, integrate their measurements, and, finally, assign weights to the indicators according to their relative importance.

Table 2.8 Summary – Integrative Framework for IS Quality Management

Model Name	An Integrative Framework for IS Quality Management
Reference	(Stylianou and Kumar 2000) ACM Citation Count: 3 (26. 05.2010) and Google : Citation Count- 28 (14.11.2010)
Origin	Developed by Stylianou and Kumar at University of North Carolina at Charlotte (US)
Purpose	Develop an integrated view of IS quality management from the viewpoint of a manager
Usage	Can be used by any IS organization to manage quality. Also, the authors

Table 2.8 Summary – Integrative Framework for IS Quality Management (continued)

<table>
<tr><td>Usage</td><td>suggest that it can be used to study the effectiveness of different TQM-based techniques in the context of IS organizations.</td></tr>
<tr><td rowspan="2">Viewpoints</td><td>No. of viewpoints: 6</td></tr>
<tr><td>Infrastructure: quality of the infrastructure (networks, systems, etc.);
Software: application software quality (development or maintenance);
Data: quality of the data from various databases;
Information: quality of the result (directly related to data quality);
Administrative: quality of management;
Service: quality of the service component (for example, help desk).</td></tr>
<tr><td>Indicators</td><td>
<table>
<tr><th>Viewpoint</th><th>Indicators and measures proposed by the authors</th></tr>
<tr><td rowspan="3">Administrative</td><td>$ per person-hour (cost)</td></tr>
<tr><td>Number of reviews
Procedures for project review and incremental commitments</td></tr>
<tr><td>Days (time – during software development)
Days (cycle time –during budgeting process)</td></tr>
<tr><td>Data, Software, and Infrastructure</td><td>Can the system interact with another system?</td></tr>
<tr><td>Information</td><td>Number of requests for help</td></tr>
<tr><td rowspan="2">Services</td><td>Survey (user satisfaction)</td></tr>
<tr><td>Response time for maintenance (quality of service)</td></tr>
</table>
</td></tr>
<tr><td>Visualization</td><td>None suggested</td></tr>
<tr><td>Notes</td><td>There are no measures proposed for data, software, or infrastructure.</td></tr>
</table>

2.1.7 The QEST & Lime Models

The QEST (Quality factor + Economic, Social, and Technical dimensions) model developed by Abran and Buglione is a performance management model specific to software engineering (Abran and Buglione, 2003; Buglione and Abran, 1999). It has the ability to handle independent sets of dimensions without predefined ratios or weights.

The initial QEST-3D model takes into consideration the following dimensions:

- Economic viewpoint: the viewpoint of managers, who are particularly interested in financial and scheduling measures taken with regard to the organizational objectives;
- Social viewpoint: the viewpoint of users, who are interested in using the software efficiently and correctly;
- Technical viewpoint: the viewpoint of developers, who are interested in the product being developed as required by the customer.

The three performance viewpoints are represented graphically as follows:

- A regular tetrahedron occupies a three-dimensional space with a regular triangular base and equal sides (pyramidal representation), as shown in Figure 2.9, and the corner EST of the triangle represents the starting point for the project: E (economic dimension), S (social dimension), and T (technical dimension) (Buglione and Abran, 1999).
- The three dimensions have the same importance with regard to overall performance, which this implies that the sides are equal and the values for every viewpoint must be normalized.
- As shown in Figure 2.9:
    - the Economic Viewpoint is represented by tetrahedron edge EP,
    - the Social Viewpoint is represented by tetrahedron edge SP,
    - the Technical Viewpoint is represented by tetrahedron edge TP;
- The value of each viewpoint is obtained by normalizing the values of the measures that have been selected to represent it.

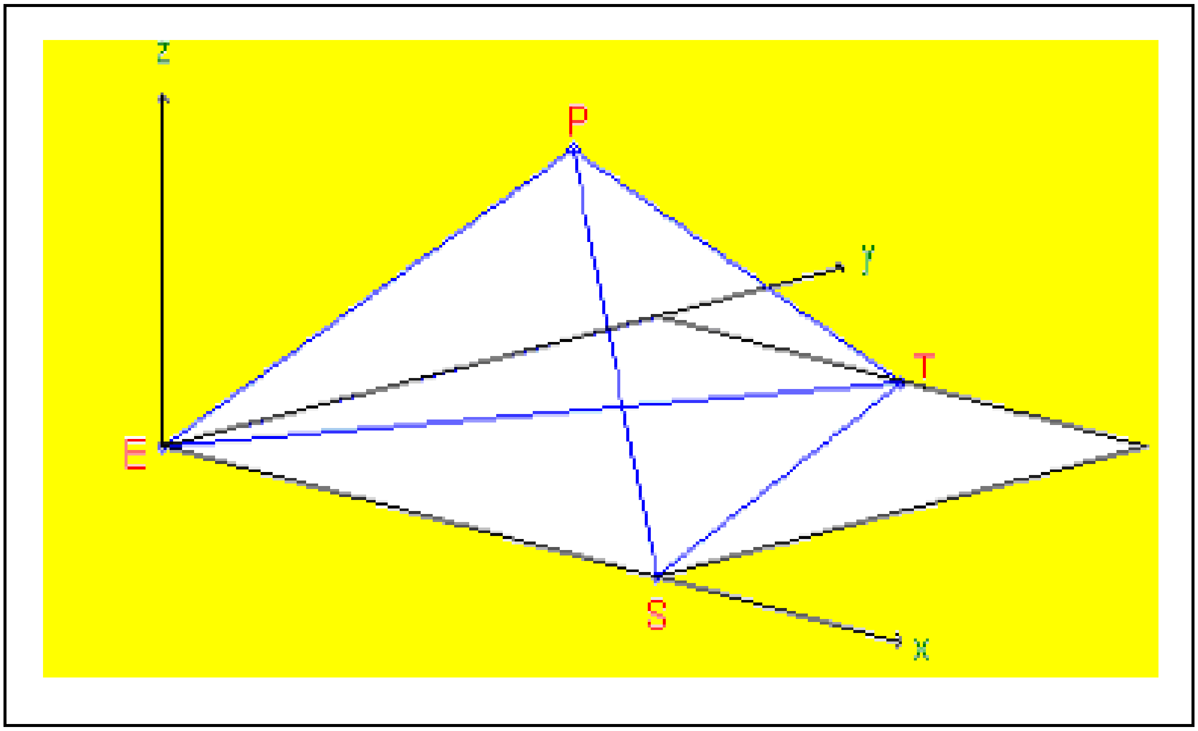


Figure 2.9 QEST model.
Adapted from Buglione and Abran (1999, p.914)
(Reprinted from Advances in Engineering Software, vol. 30 number 12, Buglione Luigi and Abran Alain, Geometrical and statistical foundations of a three-dimensional model of software performance, pp. 913-919, Copyright (1999), with permission from Elsevier.)

In this representation, the geometrical concepts of distance, area, and volume, as shown in Figure 2.10, Figure 2.11, and Figure 2.12, have the following meanings: (Buglione and Abran, 1999).

- Distances represented on the edge by EQe', SQs', and TQt' represent the advancement of the project for the three dimensions. The target coordinates to be reached are represented by (Qe, Qs, Qt) and the coordinates of the actual or current values are represented by (Qe', Qs', Qt'). At the beginning of the project, target values will be equal to zero (EQe'= EE, SQs'=SS, TQt'=TT), and at the end of the project, they are supposed to be equal to one (EQe'= EP, SQs'=SP, TQt'=TP).
- The center of the triangle represented by the initial values of the coordinates (EST) will be the starting point for the concept of distance (H) – the greater the distance between H and H', the higher the performance level will be, as shown in Figure 2.10.

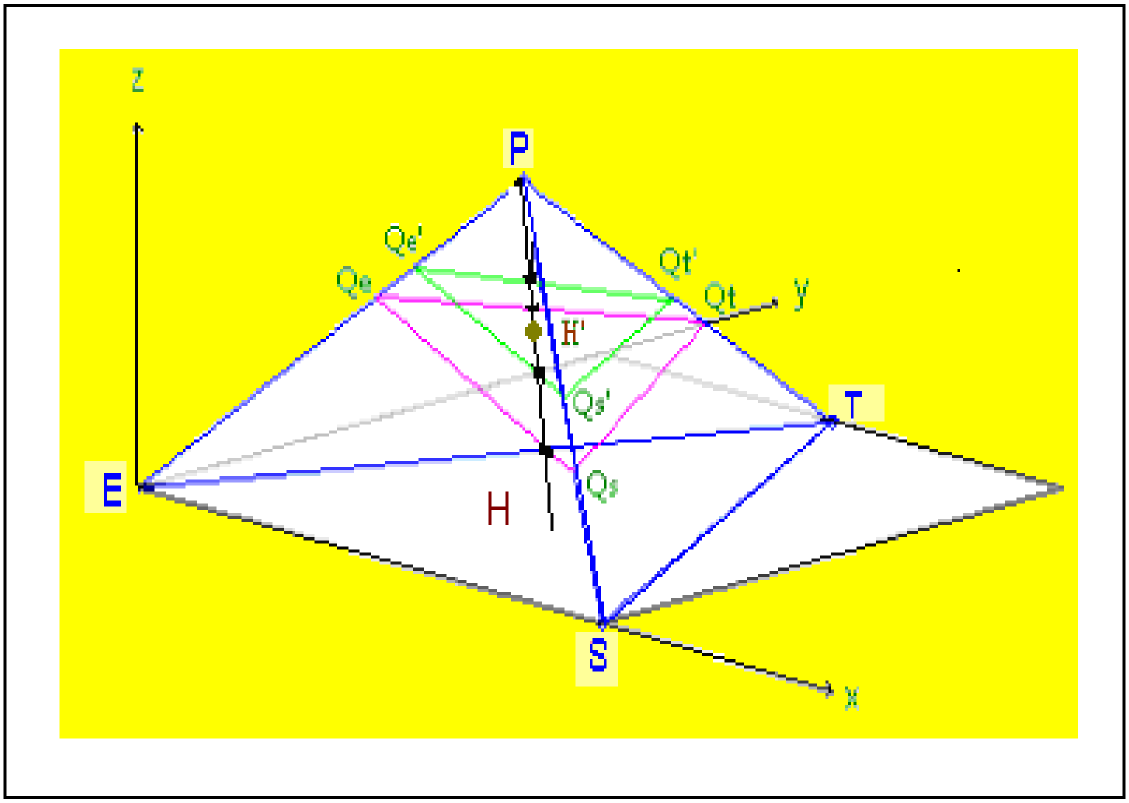


Figure 2.10 QEST model – distance between H and H'.
Adapted from Buglione and Abran (1999, p.916)
(Reprinted from Advances in Engineering Software, vol. 30 number 12, Buglione Luigi and Abran Alain, Geometrical and statistical foundations of a three-dimensional model of software performance, pp. 913-919, Copyright (1999), with permission from Elsevier.)

- Area concept: the difference between the sizes of the surfaces of the triangles given by (Qe', Qs', Qt') and (Qe, Qs, Qt), as shown in Figure 2.11 represents how performance is progressing and whether it is progressing within each viewpoint:
  - The difference will be maximum when the project is started;
  - The difference will be equal to zero when the project is finished;
  - The smaller the difference, the better the performance;
  - The section in green shows the actual performance level;
  - The triangle in red shows the target performance level;
  - The inclination angle of the section (EQe', SQs', TQt') will also give information about the progress of the project (i.e. the lowest-rated dimension or the highest-rated dimension).

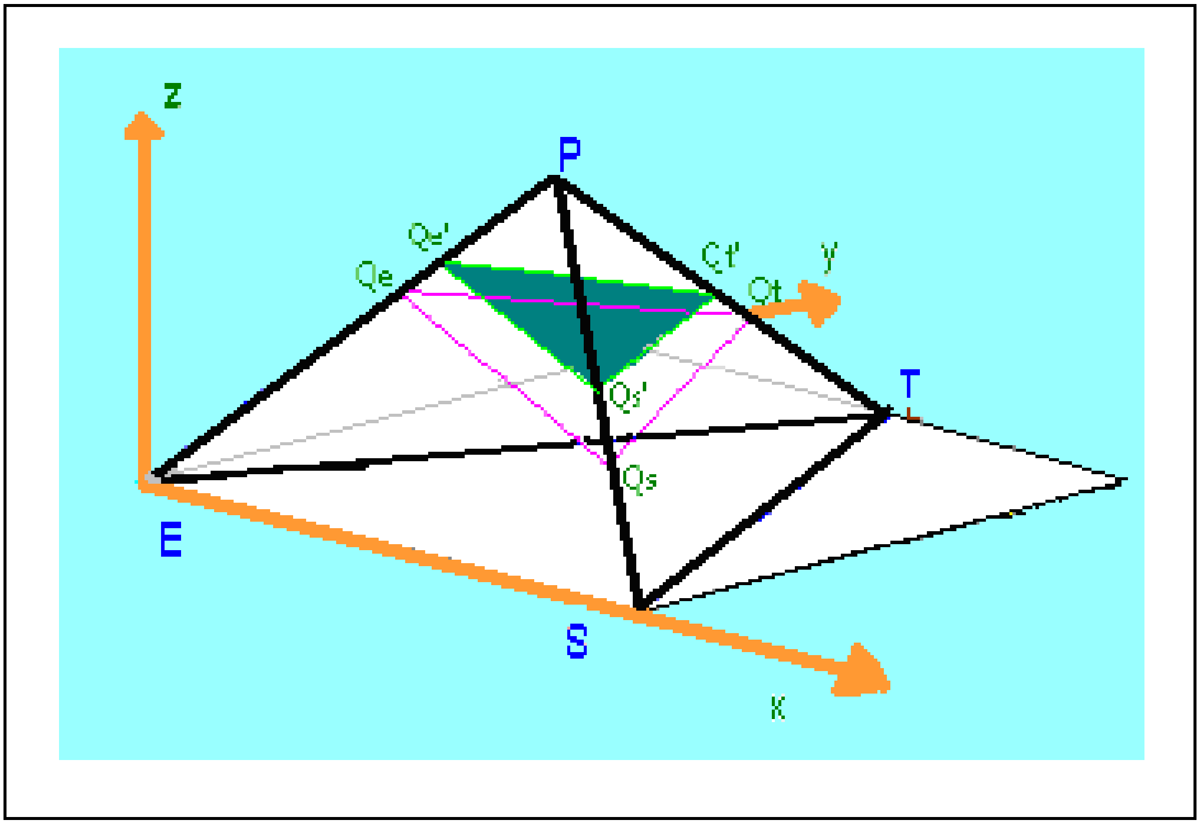


Figure 2.11 Representing performance using the geometrical concept of area.
Adapted from Buglione and Abran (1999, p.917)
(Reprinted from Advances in Engineering Software, vol. 30 number 12, Buglione Luigi and Abran Alain, Geometrical and statistical foundations of a three-dimensional model of software performance, pp. 913-919, Copyright (1999), with permission from Elsevier.)

- Volume concept: the upper part of the truncated tetrahedron shown in blue in Figure 2.12 represents the work to be done, and is delimited by the actual value given by (Qe', Qs', Qt'):
    - The smaller the upper volume, the better the performance;
    - The value of the volume will be maximum when the project is started (the performance value = 0);
    - The value of the volume will be equal to zero when the performance is maximum (project is finished);

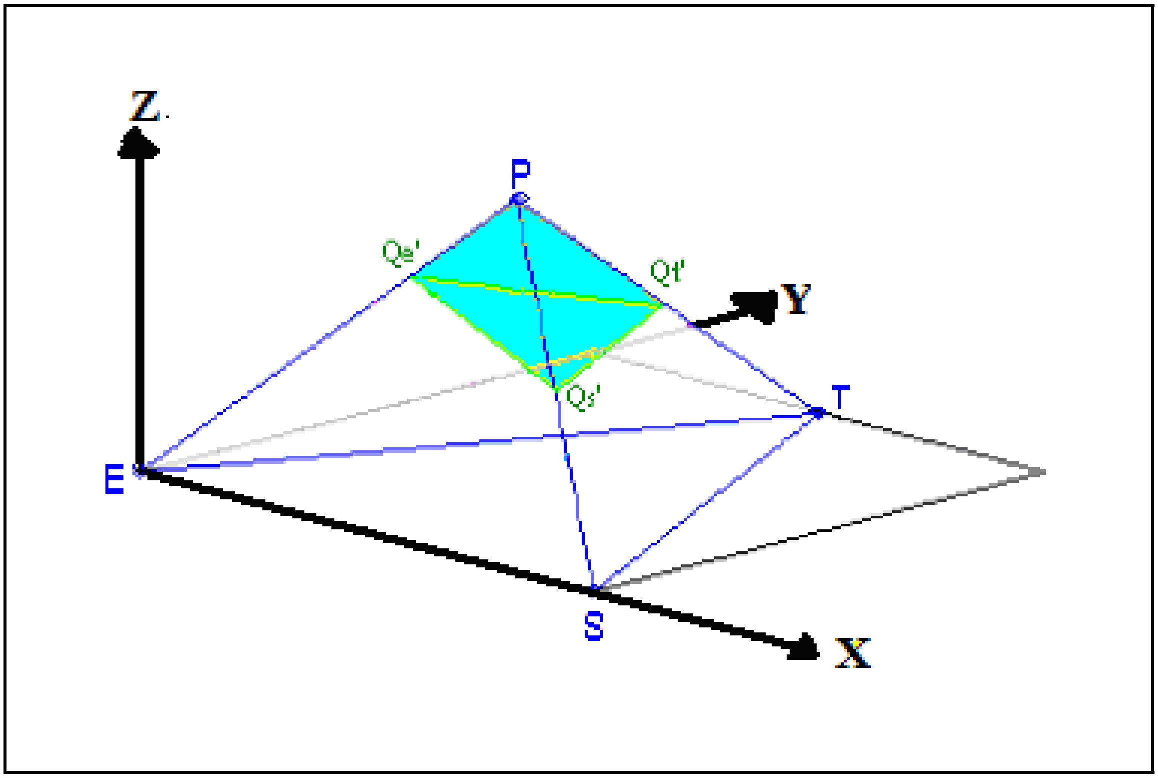


Figure 2.12 Representing performance using the geometrical concept of volume. Adapted from Buglione and Abran (1999, p.917)
(Reprinted from Advances in Engineering Software, vol. 30 number 12, Buglione Luigi and Abran Alain, Geometrical and statistical foundations of a three-dimensional model of software performance, pp. 913-919, Copyright (1999), with permission from Elsevier.)

The extension of the QEST model to *n* possible dimensions (Buglione and Abran, 2002), called QEST *n*D, is designed for more complex software projects when a larger number of dimensions needs to be taken into account. In the case of the QEST *n*D model, it is possible to handle any number of dimensions simultaneously and each viewpoint represents one of the dimensions. It cannot to be visualized for more than three dimensions.

The LIME (LIfe cycle MEasurement) model (Buglione and Abran, 2001):

- extends the QEST model concepts to make them applicable to each step of the Software Life Cycle (SLC), as managing, monitoring, and tracking performance at every SLC step is important.

- considers a generic 6-phase waterfall SLC structure (Phase 1: Requests, Phase 2: Specification, Phase 3: Design, Phase 4: Coding, Phase 5: Testing, Phase 6: Maintenance).

The suggested method for managing performance at each phase is referred to as PMAI (Plan-Measure-Act-Improve).

Table 2.9 Summary – QEST model

Model Name	The QEST model
Reference	Alain Abran and Luigi Buglione
Origin	(Abran and Buglione, 2003) Scopus Citation Count: 14 (30.10.2010) , and Google Citation Count: 38 (30.10.2010) (Buglione and Abran, 1999) Scopus Citation Count: 4 (30.10.2010), and Google Citation Count: 11 (30.10.2010) (Buglione and Abran, 2002) Scopus Citation Count: 4 (30.10.2010), and Google Citation Count: 14 (30.10.2010) (Abran *et al.*, 2003a) CiteSeer Citation Count : 1 (30.10.2010) (Buglione and Abran, 1998) Google Citation Count: 13 (30.10.2010) (Buglione and Abran, 2001) Google Citation Count: 7 (30.10.2010)
Purpose	The purpose of the QEST model is to manage the performance of software projects through a geometrical representation of performance and to provide a unitary performance value.
Usage	The QEST prototype currently available is analyzed for its strengths and weaknesses in section 2.5. There is used only in a research context to the author's knowledge.
Viewpoints	No. of viewpoints: *n*
	QEST-3D: 3 • Economic – the managers' viewpoint • Social – the users' viewpoint • Technical – the developers' viewpoint • QEST-*n*D: *n* is not predefined
Indicators	• Generic – *n* measures
Visualization	• The pyramidal concept is easy to understand by managers using three dimensions. • What happens when there are more than three dimensions? The proposed geometrical representation is a simplex, but it cannot to be visualized for more than three dimensions. • There is no existing tool that supplies a visual representation for the QEST-nD model, and the only prototype developed to use QEST is limited to three viewpoints, as shown in section 2.5.

Table 2.9 Summary – QEST model (continued)

Notes	• This open model (meaning that it can be filled in according to management objectives) could be used as a complement to the BSC (Abran and Buglione, 2003). • The performance values are consolidated using a single performance index. • Indicators and measures (characteristics and sub-characteristics) from ISO 9126 could be analyzed using the QEST model.

2.2 Generic Performance Management Frameworks

This section presents a synthesis of the literature related to PMFs in the generic field of management. Over the years, several frameworks have been developed to address the management of organizational assets, both tangible and intangible.

In recent decades, generic PMFs have emerged using various approaches:

- The BSC, which manages strategy with four viewpoints combining financial and non financial measures (Kaplan and Norton, 1992; 1993; 1996a; 1996b). It was included in this literature review because it is one of the most influential PMFs currently available.
- The Performance PRISM (Neely and Adams, 2005b; Neely *et al.*, 2002), which was selected because it is easy to visualize. It was developed by the Centre for Business Performance at the Cranfield School of Management, a research center in performance management.
- The EFQM Excellence Model (European Foundation for Quality Management) (European Foundation for Quality Management, 2008) and Baldrige Criteria for Performance Excellence (Baldrige, 2005; English, 1991). These are well known excellence models designed to be used in a systematic and structured way, and are well accepted.
- Skandia and IAM (Sveiby, 1997a; 1997b; 2004). These models are well known for their use of intangible measures.

- The Integrated Performance Measurement System (IPMS) (Laitinen, 2002). This model was designed in Finland for smaller organizations.
- Sink and Tuttle (Sink, 1985; Sink and Tuttle, 1989). This is one of the first ever PMFs.

2.2.1 Balanced Scorecard (BSC)

In response to the shortcomings of traditional financial data for evaluating organizational performance, a new framework was developed in 1992, known as the Balanced Scorecard (Kaplan and Norton, 1992). From its beginnings as a Performance Measurement Framework, the BSC has evolved into a PMF, and, according to Niven, "so widely accepted and effective...that the Harvard Business Review recently hailed it as one of the 75 most influential ideas of the twentieth century" (Niven, 2002, p. 12).

The BSC is used to evaluate corporate performance from four different viewpoints: financial, internal business process, customer, and learning and growth, and can be adapted to measure IT performance.

Neely (Neely, 2005) investigated the most often cited references on the subject of the Performance Measurement Frameworks between 1995-2004, and the conclusion was that the most cited reference in 1995 and from 1998 to 2004 is (Kaplan and Norton, 1992).

Kaplan and Norton founded the Balanced Scorecard Collaborative, Inc. (BSCol) to "facilitate the worldwide awareness, use, enhancement, and integrity of the Balanced Scorecard as a value-added management process" (Morisawa, 2002). According to Martinsons *et al. (Martinsons et al., 1999)*, a general BSC could be used either at the departmental level or at the software application level.

The BSC has become popular, notably for Japanese companies, and one example at Kansai Electric Power includes the strategy map, strategic communication, strategy focus, and change in the organizational climate (Morisawa, 2002). KPMG's performance measurement white paper (KPMG, 2001) outlines several drawbacks to Kaplan and Norton's BSC. Among them is the contention that the four viewpoints are too limiting, notably because there is a lack of consideration in the existing viewpoints for knowledge creation processes and intellectual capital. A second criticism is that there is little focus on the external environment. Kaplan and Norton argue that these:

> "four perspectives should be considered as a template, not as a straight jacket. No mathematical theorem exists that four perspectives are necessary and sufficient. We have yet to see companies using fewer than these four perspectives, but, depending on industry circumstances and a business unit's strategy, one or more additional perspectives may be needed" (Kaplan and Norton, 1996a, p. 34).

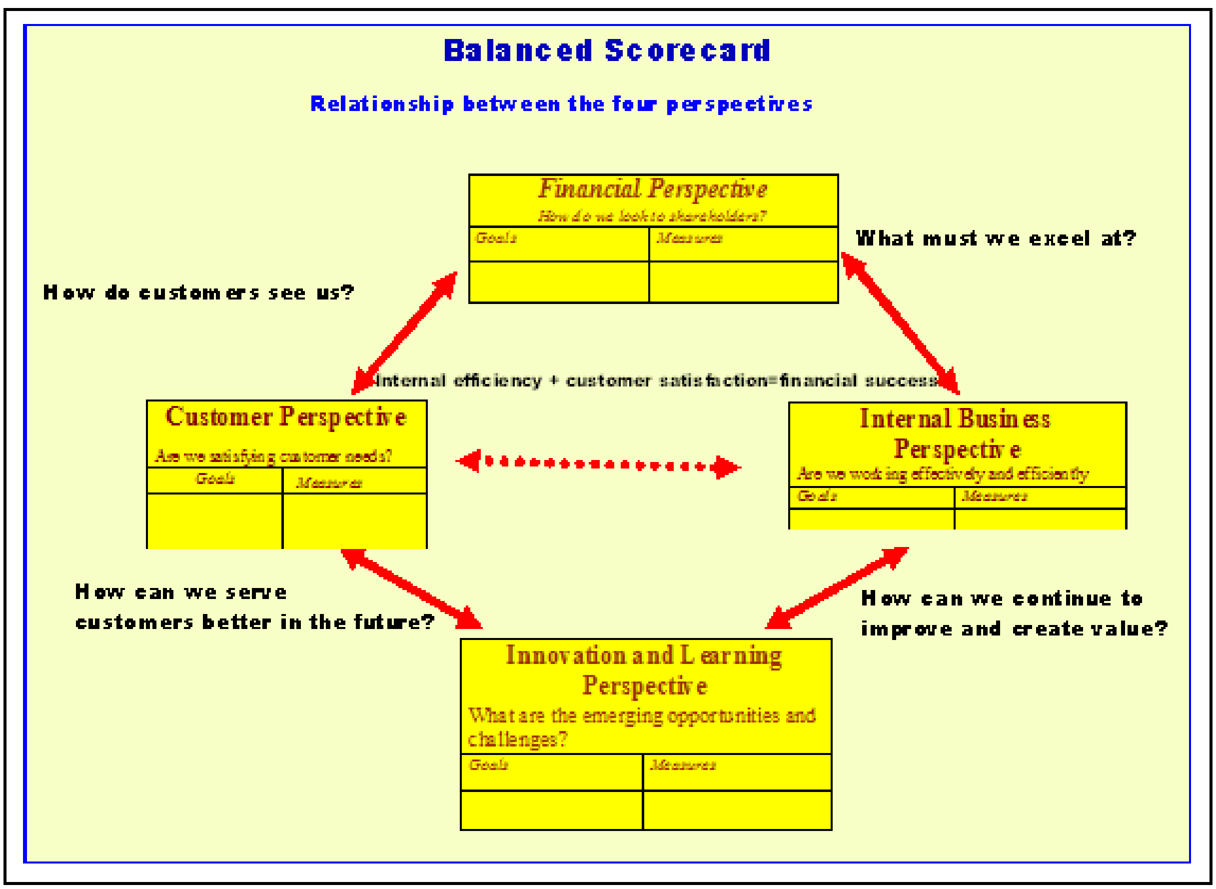


Figure 2.13 Relationship between the four viewpoints in BSC.
Adapted from Kaplan and Norton(1998, p.136)
(Reprinted from The Economic Impact of Knowledge, Robert S. Kaplan, David P. Norton, chapter 18, Putting the Balanced Scorecard to Work, p. 10, Copyright (1998), with permission from Elsevier.)

Table 2.10 Summary – Balanced Scorecard

Model Name	Balance Scorecard (BSC)
Reference	(Kaplan and Norton, 1996b) Google : Citation Count- 2549 (14.11.2010) (Kaplan and Norton, 1993) Google : Citation Count- 1582 (14.11.2010) (Kaplan and Norton, 1992) Google : Citation Count- 8431 (14.11.2010) (Kaplan and Norton, 1996a)
Origin	Developed by Robert Kaplan and David Norton in 1992 at the Nolan Norton Institute.
Purpose	The purpose of the BSC is to manage organizational performance at all levels.
Usage	Adopted by more than 50% of the largest U.S. organizations (Marr and Neely, 2003). The Army & Air Force Systems Service (AAFES), a $9 billion global retailer with 50,000 employees serving 8.7 million customers in 3,100 stores in 30 countries, uses the BSC model, and, for a period of four years, "revenue increased by 11%, dividends by 19%, employee satisfaction by 16%, and customer satisfaction by 17%. Finally, inventory was reduced by $108 million" (Scorecard, 2009).
Viewpoints	No. of viewpoints: 4
	The four original viewpoints for the BSC are: Financial, Internal Business Process, Customer, and Learning and Growth. **Financial Viewpoint**: should indicate whether or not the strategy, implementation, and execution have a role in bottom-line improvement (Kaplan and Norton, 1996a, p. 25). **Customer Viewpoint**: identifies market segments in which the organization market will compete and the target measures for this market segment (Kaplan and Norton, 1996a, p. 26). **Internal Business Process Viewpoint**: identifies the most critical processes and incorporates innovation processes into the internal business processes, as shown in (Kaplan and Norton, 1996a, pp. 27-28). **Learning and Growth**: identifies the infrastructure that organizations must build to create long-term growth and improvement. There are three principal resources for this viewpoint: people, systems, and organizational procedures and measures must be developed for each measure (Kaplan and Norton, 1996a, pp. 28-29).
Indicators	The proposed generic measures are (Kaplan and Norton, 1996a, p. 44): **Financial**: return on investment and economic value added **Customer**: satisfaction, retention, market, and account share **Internal**: quality, response time, cost, and new product introduction **Learning and Growth**: employee satisfaction and information system availability
Visualization	None suggested by the authors and there is no standard tool to represent BSC.

2.2.2 The Performance Prism

The Performance Prism measurement framework was developed by the Centre for Business Performance at the Cranfield School of Management in cooperation with the Process Excellence Core Capability Group of Andersen Consulting.

The Performance Prism is described using five viewpoints: Stakeholder Satisfaction, Stakeholder Contribution, Strategies, Processes, and Capabilities (as shown in Figure 2.14). These five viewpoints on performance can be geometrically represented in the form of a prism.

DHL, a well known international express courier company, is one of the first users of the Performance Prism. Managers use it to establish what should be discussed at their quarterly performance reviews and also for the day-to-day tracking of performance (Neely and Adams, 2003).

According to the author, “an organization’s results (stakeholder satisfaction) are a function of determinants (the other prism facets)…the framework is comprehensive, enabling all measures to be mapped on to it so the gaps in measurement can be identified” (Neely, 2002, p. 152).

Below is a description of how the five viewpoints of performance are represented graphically. Each of the Performance Prism's five interrelated facets, as shown in Figure 2.14 and Table 2.11, represents a key area that determines success. The weight given to each facet will depend on the particular strategic objectives. The surface areas are directly related to the weight given to each objective.

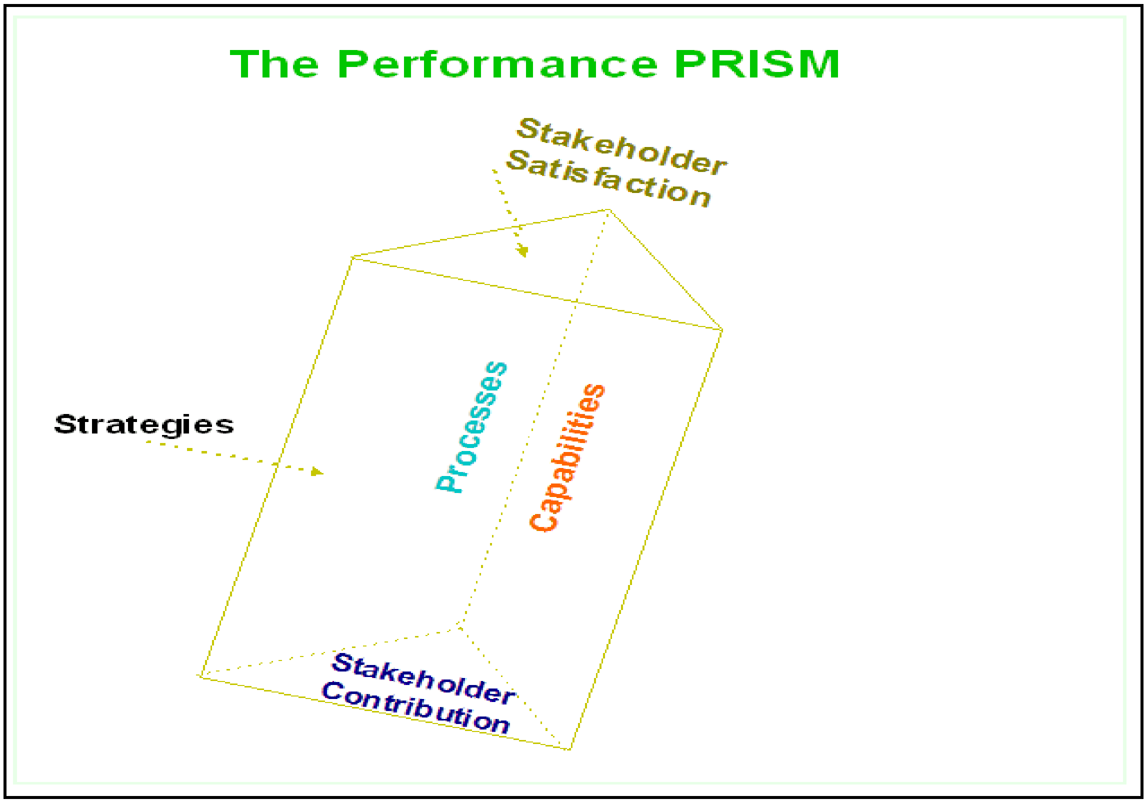


Figure 2.14 The five dimensions of the performance prism.
Adapted from Neely and Adams (2005a, p.41-48)

Table 2.11 Summary –Performance Prism

Model Name	The Performance Prism
Reference	(Neely, 2002) Google : Citation Count- 150 (11.11.2010) (Neely, 2005) Google : Citation Count- 86 (11.11.2010) (Neely *et al.*, 1996) Google : Citation Count- 65 (11.11.2010) (Neely *et al.*, 2001) Google : Citation Count- 106 (11.11.2010) (Adams and Neely, 2005; Neely and Adams, 2003; 2005a; 2005b)
Origin	Developed by a number of groups of people via a series of workshop discussions over the years since 1998. Developed at Andersen Consulting (now Accenture) – Centre for Business Performance at the Cranfield School of Management (formerly at The Judge Institute, Cambridge University)
Purpose	Manage organizational performance adopting a three-dimensional visualization in the form of a prism called the Performance Prism

Table 2.11 Summary –Performance Prism (continued)

Usage	According to (Neely and Adams, 2003), DHL is using the Performance Prism.
Viewpoints	No. of viewpoints: 5
	The definitions of the viewpoints or dimensions are as follows: • **Stakeholder Satisfaction**: measure what is most important to achieve it; • **Stakeholder Contribution**: measure what is required for stakeholders to maintain and develop the organization; • **Strategies**: measure what is important to put in place to successfully develop them; • **Processes:** measure them to determine which to put in place to achieve the strategy; • **Capabilities**: measure those that are necessary to operate and develop processes.
Indicators	• Over 200 measures have been developed to be used as a guide (Neely, 2002, p. 153).
Visualization	Geometrical representation of a prism, as shown in Figure 2.14, makes it possible to easily visualize internal measures (strategy, capability, and process) and external measures (stakeholder).
Notes	• The focus is clearly on the stakeholders. ▪ The five viewpoints are distinct, but at the same time are logically linked through the Prism representation.

2.2.3 EFQM Excellence Model

The EFQM Excellence Model (European Foundation for Quality Management, 2008) is a non prescriptive self-assessment approach developed by 14 multinationals that belong to the European Foundation for Quality Management. A history of the EFQM model and a comparison of this model with the TQM (Total Quality Management) model can be found in (Conti, 2007).

According to Mavroidis *et al.*, EFQM is the most widespread framework to be developed in the last decade in Europe (Mavroidis *et al.*, 2007), and is used in assessing applications to the European Quality Award. It uses nine viewpoints, covering leadership; policy and strategy; people; partnerships and resources; processes; customer results; people results; society results; and key performance results, as shown in Figure 2.15 and Table 2.12. The main purpose of this model was to make it the reference model for the promotion and presentation of the European Award for Quality Management.

Table 2.12 Viewpoints (Criteria) for EFQM
Taken from (British Quality Foundation, 2005)

Viewpoints	Questions
1. Leadership	• How are the resources planned, managed, and improved? • How does the manager develop a mission and a vision? • How are the managers involved in ensuring that the organization's management system is developed, implemented, and improved? • What is the state of the interaction with customers, partners, and societal representatives? • How do managers identify and champion organizational change and reinforce a culture of excellence?
2. People	• How are the resources planned, managed, and improved? • How are knowledge and competencies identified, developed, and sustained?
3. Policy and Strategy	• Are they based on current and future needs and on the expectations of stakeholders? • Are they based on information from performance measurement, research, learning, and related external activities? • Are they developed, reviewed, updated, and communicated?
4. Partnerships & Resources	• How are finances managed? • How are buildings, equipment, and materials managed? • How is technology managed? • How are information and knowledge managed?
5. Process	• How are processes improved to increase value for customers and other stakeholders? • How are products and services adapted to customer needs and expectations? • How are products and services produced, delivered, and serviced? • How are customer relationships managed and enhanced?
6. People	• What is achieved in relation to its employees?
7. Customers	• What customer expectations are fulfilled?
8. Society	• What is achieved in terms of satisfying the needs and expectations of society?
9. Key Performance	• What has the organization achieved, as measured by comparing real performance with target performance using Key Performance Outcomes and Key Performance Indicators

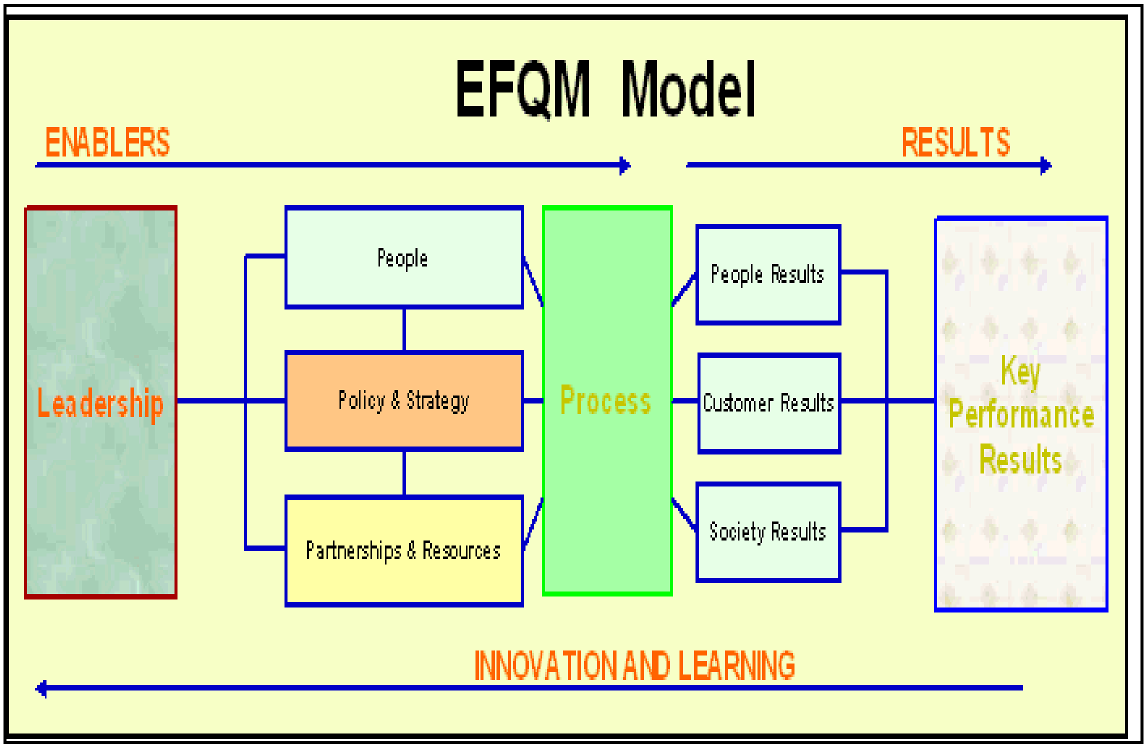


Figure 2.15 EFQM model
Adapted from The British Quality Foundation (2005)

Table 2.13 Summary –EFQM model

Model Name	The EFQM Framework Its original name was the European Foundation for Quality Management model, and is now referred to as the EFQM Excellence Model.
Reference	(European Foundation for Quality Management, 2008; George *et al.*, 2003; Mavroidis *et al.*, 2007; Porter and Tanner, 2004) (Conti, 2007) Google : Citation Count- 19 (07.11.2010) (Westerveld, 2003) Scopus Citation Count- 40 (15.11.2010) (British Quality Foundation, 2005)
Origin	The EFQM was officially established in 1989. The European Quality Award (EQA) is based on the framework of the EFQM.
Purpose	Its purpose is to manage organizational quality and performance across European organizations to help them in their drive to becoming more competitive.

Table 2.13 Summary –EFQM model (continued)

Usage	• Its applicability in UK universities is discussed by Davies (Davies, 2008). • In Europe, more than "20,000 organizations currently using [sic: use] the model to drive their improvement activities" (Porter and Tanner, 2004, p. 151).
Viewpoints	No. of viewpoints: 9
	The model is based on 9 viewpoints, 4 of them classified as 'Results' and 5 of them as 'Enablers', as shown in Figure 2.15 and explained in . • 4 'Results' : People, Customers, Society, and Key Performance Results. • 5 'Enablers' : Leadership, People, Policy & Strategy, Partnerships and Resources, and Process.
Indicators	Every organization has its own measures for evaluating performance based on the 9 viewpoints explained in Table 2.12.
Visualization	• No visualization approaches or techniques are suggested.
Notes	▪ Possibility of benchmarking (comparison) with a broad diversity of other organizations ▪ Viewpoints and indicators of the EFQM reviewed and improved annually

2.2.4 Intangible Assets Monitor (IAM)

The Intangible Assets Monitor (IAM), as shown in Figure 2.17 and Figure 2.16, is a method for measuring intangible assets. According to Sveiby, the choice of indicators depends on the company's strategy (Sveiby, 1997a; 1997b). There are financial (visible and invisible) indicators and non financial (tangible and intangible) indicators, as shown in Figure 2.17. All the assets (tangible or intangible) of an organization are the result of human actions (Sveiby, 1997b, p. 8).

The overall rating is achieved by generating an index from each indicator and giving all indicators an equal weight (unweighted average). Human actions are converted into intangible assets and tangible indicators: the first dealing with external, internal, and competence indicators, and the second dealing with growth, renewal, efficiency, and stability.

The claim of this model is that all intangible assets can be classified into three viewpoints: external structure, internal structure, and individual competence, as shown in Figure 2.17, Figure 2.16, and Table 2.14.

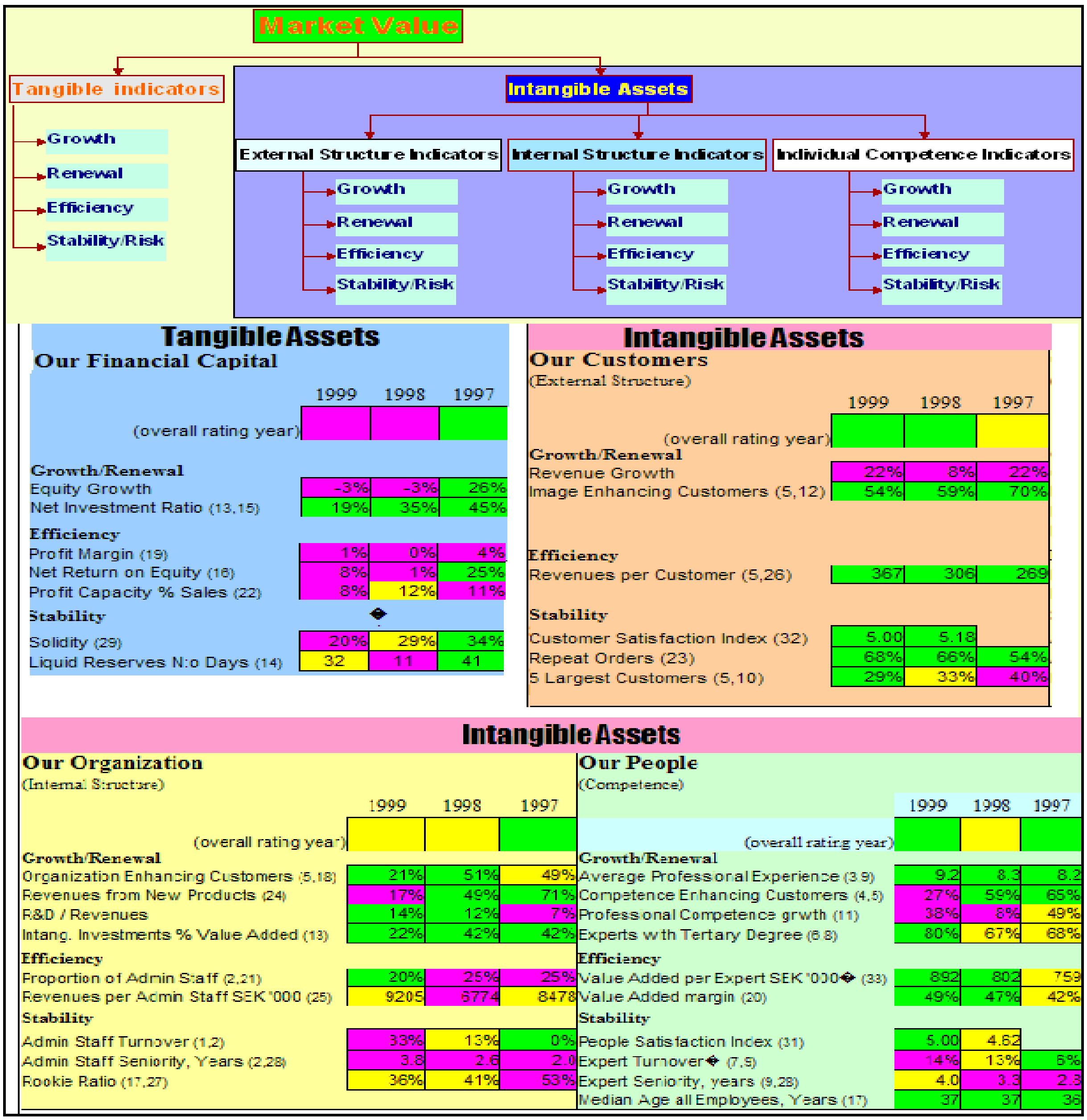


Figure 2.16 Intangible assets monitor
Adapted from Sveiby (1997a) and Monitor (1999)

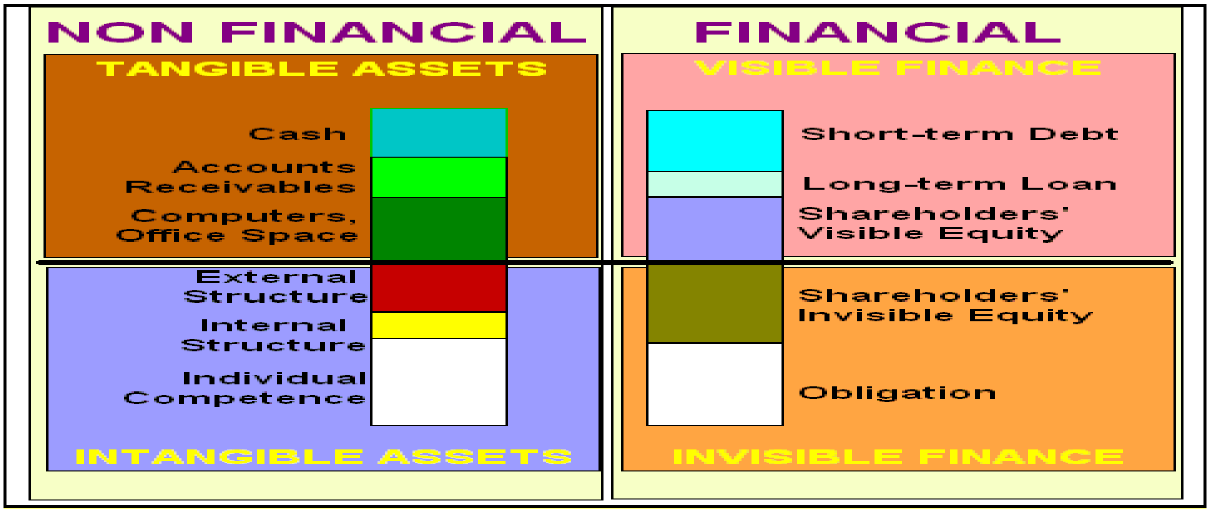


Figure 2.17 The balance sheet of a knowledge organization
Adapted from Sveiby (1997b, p.11)

Table 2.14 Table of measures
Adapted from from Sveiby (1997b, pp.12,195)

	External Structure Indicators (These indicators, about brands, customer and supplier relationships, are important in helping organizations assess their potential for development.)	**Internal Structure Indicators** (These indicators are concerned with the organization's management and legal structure; manual systems, attitudes, R&D, software and the resources involved in maintaining these structures.)	**Competence Indicators** (These indicators are concerned with education and experience, meaning the competence to act in such a way as to create both intangible and tangible assets in a wide variety of situations.)
Indicators of growth/renewal	• Profitability per customer • Image-enhancement for customers	• Investment in IT • Structure-enhancement for customers	• Number of years in the profession • Level of education • Training and educational costs • Competence-enhancement for the customer • Total competence of experts (years)

Table 2.14 Table of measures (continued)

	External Structure Indicators (These indicators, about brands, customer and supplier relationships, are important in helping organizations assess their potential for development.)	**Internal Structure Indicators** (These indicators are concerned with the organization's management and legal structure; manual systems, attitudes, R&D, software and the resources involved in maintaining these structures.)	**Competence Indicators** (These indicators are concerned with education and experience, meaning the competence to act in such a way as to create both intangible and tangible assets in a wide variety of situations.)
Indicators of efficiency	• Satisfied customer index • Sales per customer • Win/loss index	How much value do the employees produce? • Proportion of support staff • Value/attitude index	• Proportion of professionals • Value added per employee • Value added per professional • Profit per employee • Profit per professional
Indicators of stability&risk	• Proportion of large customers • Devoted customer ratio • Frequency of repeat orders	• Age of the organization • Support staff turnover • Rookie ratio • Administration staff seniority (years)	• Professional turnover • Expert seniority (years) • Median age of all employees (years)

Table 2.15 Summary – IAM model

Model Name	Intangible Assets Monitor (IAM)
Reference	(Sveiby, 1997a; 2001; 2004) (Sveiby, 1997b) Google : Citation Count- 1944 (18.11.2010)
Origin	According to Sveiby, the model was developed using data from some 40 knowledge management initiatives taken by organizations and practitioners (Sveiby, 2001).
Purpose	The purpose of the IAM model is to measure and manage intangible assets.
Usage	• The first Swedish organization to use this concept was Celemi in 1995 (Sveiby, 1997b, p. 191).

Table 2.15 Summary – IAM model (continued)

Viewpoints	5
	• Market value has 2 viewpoints: tangible net book value and intangible assets • An intangible asset has three viewpoints: external structure, internal structure, and individual competence, as shown in Table 2.14 and Figure 2.16.
Indicators	• Efficiency, stability/risk, and growth/renewal, as shown in Table 2.14 and Figure 2.16.
Visualization	As shown in Figure 2.16, using the Celemi Monitor (Monitor, 1999): If higher values are considered better: ▪ the cells are colored green – their value is equal to or greater than the target. Red cells indicate that their value is 80% less than the target. Yellow cells indicate values in between. If lower values are considered better: ▪ the cells are colored green – their value is equal to or less than the target. Red cells indicate that the value is 20% higher than the target. Yellow cells indicate values in between. The viewpoints included are determined in advance and all indicators have equal weights (unweighted average).
Notes	▪ This method contributed to the development of Skandia's Business Navigator, which will be described in section 2.2.6. ▪ It is important to mention that the same color represents different performances, as a function of whether or not the values are considered better (lower or higher). Otherwise, the visualization approach adopted is very similar to a dashboard: easy to understand and interpret. ▪ The model could be used at the organizational level and at the individual level (Sveiby, 2001), as well to manage intangible assets.

2.2.5 The Baldrige Framework

The Baldrige Award (Baldrige, 2005) provides a comprehensive framework for performance measurement which can be used by organizations in these categories: manufacturing businesses, service businesses, small businesses, health care organizations, and educational organizations (Brown, 2001).

An organization can apply for the Baldrige Award and use the feedback as an assessment tool (Srivivatanakul and Kleiner, 1996). Brown explains and details the criteria for the 2003 release (Brown, 2003). The relationship between the viewpoints is shown in Figure 2.18 (Evans, 2004), where a box represents one viewpoint. The 7 viewpoints of the model contain 24 key focus items and 52 specific areas for improvement.

Initially, this model was used for quality assessment. Then, in 1995, the word 'quality' was removed from the list of criteria. The new criteria have a more balanced approach, and the introduction of the business results viewpoint constitutes an important step in managing organizational performance.

There is a guide available, as well as other literature, to help organizations understand and analyze the measures, and shows them how to use these criteria (Brown, 2001; 2003).

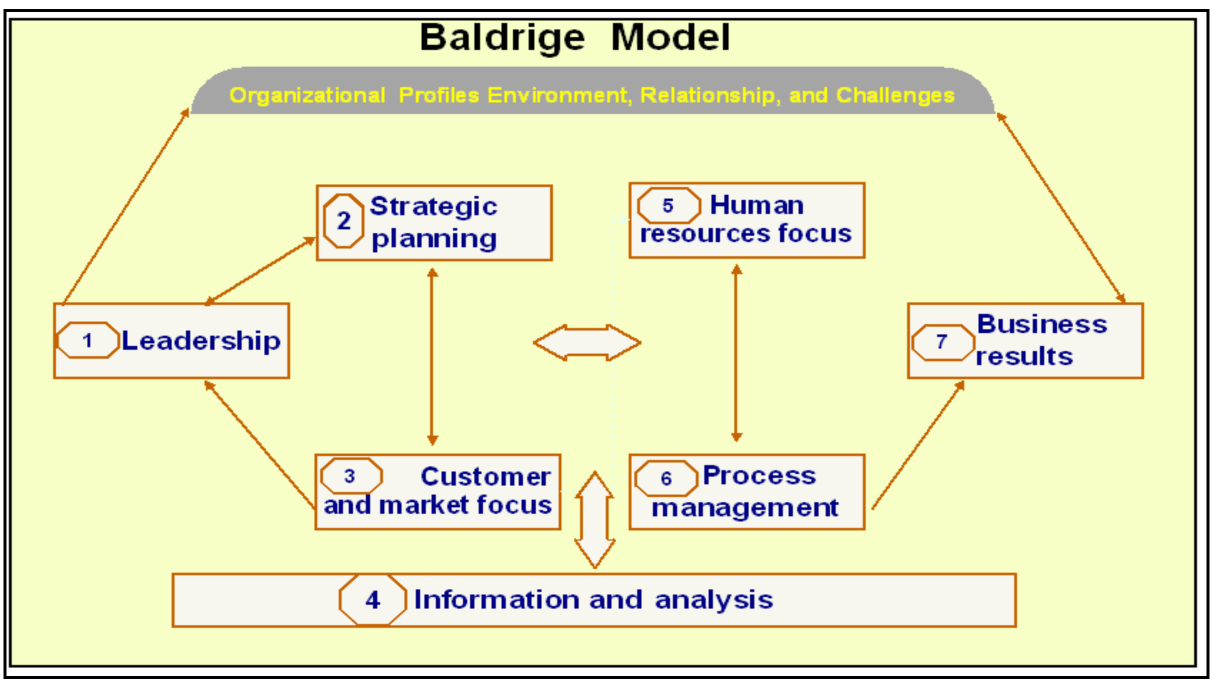


Figure 2.18 Baldrige model.
Adapted from Evans (2004, p.220)
(Reprinted from Journal of Operations Management, vol. 22, James R. Evans, An exploratory study of performance measurement systems and relationships with performance results, p. 14, Copyright (2004), with permission from Elsevier.)

Table 2.16 Summary – Baldrige model

Model Name	The Baldrige Framework
Reference	(Baldrige, 2005; 2010; Brown, 2001; Srivivatanakul and Kleiner, 1996; Tudahl and Lindner, 1994); (Liebesman, 1991); (Flynn and Saladin, 2001) Google : Citation Count- 82 (19.11.2010) and Scopus Citation Count - 54 (19.11.2010) ; (Brown, 2003) Amazon.com: 15 books cite this work (17.11.2010)
Origin	It was launched by the U.S. government in 1987. The existence of the Baldrige Award is based on Public Law 100–107. A maximum of 9 organizations can be winners each year: 3 large manufacturing companies, 3 large service companies, and 3 small businesses. This is a public-private partnership with the goal of encouraging quality (Brown, 2001).
Purpose	The purpose is to manage organizational quality and performance. According to Brown, the overall purpose is "to strengthen the competitiveness of U.S. companies" (Brown, 2001). Other roles are: "To help improve organizational performance… To facilitate communication and sharing of best practices information… To serve as a working tool for understanding and managing performance…" (Brown, 2001).
Usage	• Could be applied in a service or manufacturing organization, and a small or large organization; • Recognized and "used by thousands of organizations around the world to evaluate their progress toward becoming the best in their fields" (Brown 2009, p. V). • "Several million copies of the criteria have been distributed to organizations all over the world" (Brown 2009, p. 2).
Viewpoints	No. of viewpoints: 7
	• **Leadership:** will show how top-level managers guide all the activities, both within and outside the organization, that support development; • **Strategic planning:** will show the manner in which strategic objectives are set and how they are put into practice; • **Customer and market focus:** will show the manner in which requirements and expectations are fixed and how relationships with customers and other organizations are established; • **Information and analysis:** will show how effective the use of data and information is in supporting key organizational processes and the organization's performance management system. Emphasis is notably placed on how the organization selects, collects, and analyzes measures to assess organizational performance. • **Human resources focus:** examines how the organization enables its workforce to develop and become aligned with the organization's objectives; • **Process management:** examines how processes are designed and effectively managed in order to improve performance;

Table 2.16 Summary – Baldrige model (continued)

Viewpoints	• **Business results:** the most important dimension, contributing to 45% of the score, showing the organization's performance: customer satisfaction, financial and marketplace performance, human resources, supplier and partner performance, operational performance, and governance and social responsibility;
Indicators	According to Brown, there are 18 examination items and 32 areas to address, and: "While each of the seven Categories is evaluated separately, there are relationships (or 'linkages') between the seven and they function together as a system" (Brown, 2003).
Visualization	• There is no a standard tool for this framework.

2.2.6 Skandia Navigator

Skandia (a Fortune 500 company) (Skandia, 1994), founded in 1855 as a property and insurance company, is now an insurance and financial services organization in Sweden with offices in more than 20 countries. They published a supplement to their 1994 annual report on visualizing intellectual capital in Skandia (Skandia, 1994). Their framework, which they call Skandia Navigator, enables them to manage intellectual assets by using a collection of critical measurements aimed at providing a balanced view of performance and goal achievement, as shown in Figure 2.19. The concept is close to that of Kaplan and Norton's BSC, which includes financial and non financial viewpoints. It is based on the identification of critical indicators organized into five viewpoints linked to the value creation process: a financial focus, a customer focus, a process focus, a renewal and development focus, and a human focus. A fifth viewpoint on human resources has therefore been added to the original four viewpoints found in the BSC.

A second-generation model was also produced, which is aimed at consolidating all the various individual indicators into a single index and at correlating the changes in intellectual capital (IC) with changes in market value. Every year since 1994, Skandia has published a supplement to their annual report (Skandia, 2006).

It is not easy to manage IC, and the selection of intangible measures is poorly understood, so the advantage of this model is to facilitate IC analysis using a logical link between measures. The term ‘intellectual capital’ is defined as “the possession of knowledge, applied experience, organizational technology, customer relationships, and professional skills” (Edvinsson, 1997). Managing IC “provides Skandia AFS with a competitive edge in the market” (Edvinsson, 1997).

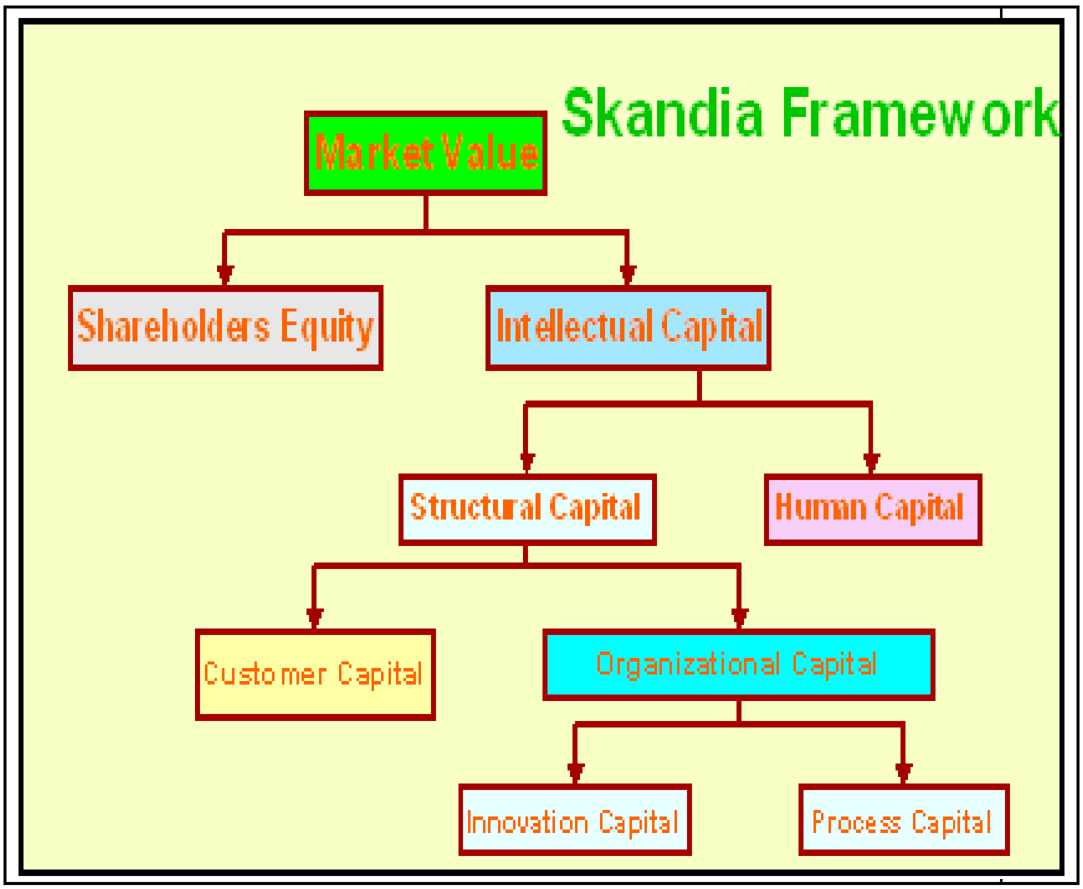


Figure 2.19 Skandia framework.
Adapted from Edvinsson (1997, p.369)
(Reprinted from Elsevier Science, vol.30, Edvinsson, Leif, Developing intellectual capital at Skandia, pp. 366-373, Copyright (1997), with permission from Elsevier.)

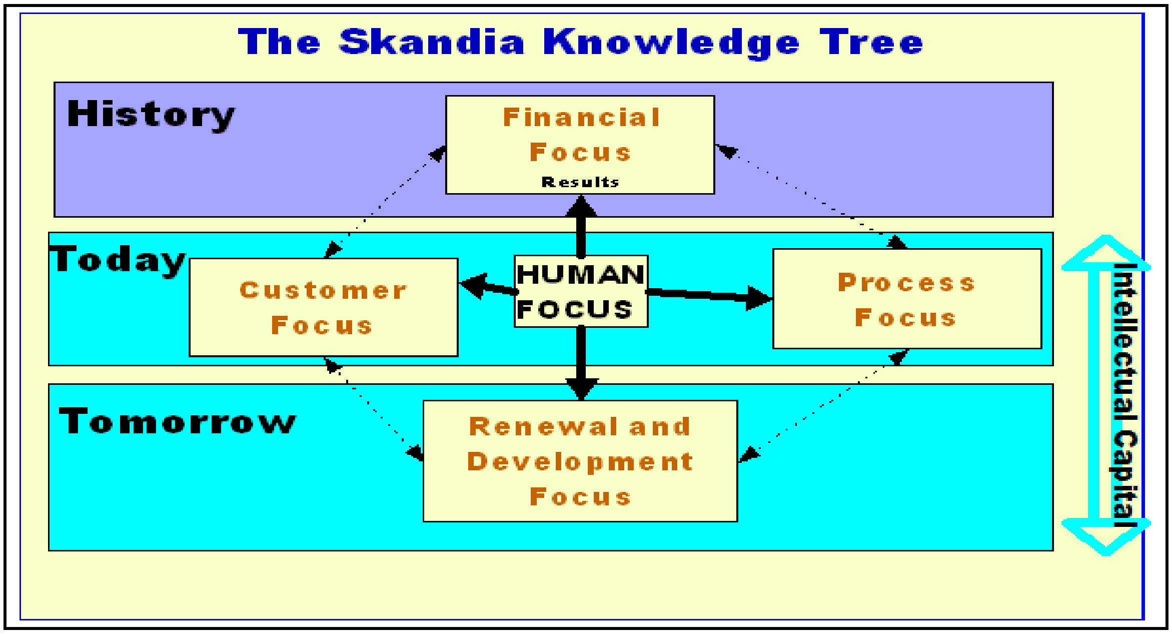


Figure 2.20 Skandia knowledge tree.
Adapted from Edvinsson (1997, p.371)
(Reprinted from Elsevier Science, vol. 30, Edvinsson, Leif, Developing intellectual capital at Skandia, pp. 366-373, Copyright (1997), with permission from Elsevier.)

Table 2.17 Summary – Skandia model

Model Name	The Skandia Model
Reference	(Skandia, 1994; 2006; Sveiby, 2004) (Edvinsson and Malone, 1997) Google : Citation Count- 1454 (16.11.2010) (Edvinsson, 1997) Scopus Citation Count- 144 (19.11.2010) (Edvinsson, 2000) Google : Citation Count- 60(16.11.2010)
Origin	ICM Skandia – The first balanced report was created in 1994 and made publicly available starting in 1995 (Edvinsson, 1997)
Purpose	The purpose is to calculate performance for intangible organizational capital by defining IC (Intellectual Capital) as Human Capital + Structural Capital.
Usage	Skandia (Skandia, 1994)
Viewpoints	No. of viewpoints: 5
	The value of intellectual capital is derived via: **Process focus** – indicators related to the actual process of creating the services and products. **Financial focus** – indicators related to the financial outcomes of the organization; **Customer focus** – indicators related to how well the organization meets the needs of its customer; **Renewal and development focus** – indicators related to the long-term

Table 2.17 Summary – Skandia model (continued)

Viewpoints	renewal of the organization; **Human focus** – indicators related to the process of knowledge creation, probably the most important for creating value.
Indicators	IC value is calculated through the use of 164 indicators (Edvinsson, 1997, p. 57) divided along the five viewpoints (Balantzian, 2005, p. 210). According to Li and Ke-Yi, "there are 113 measurement indicators of IC elements in Skandia Navigator, but many of them are non-public information" (Li and Ke-yi, 2009); Examples of some key indicators are: **Process focus**: administrative expenses/managed assets, and cost of administrative errors/management revenues; **Financial focus**: fund assets (millions of Swedish krona (MSEK)), fund assets/employee (MSEK), and income/managed assets; **Customer focus**: market share, number of accounts, and customers lost **Renewal and development focus**: competency development expenses/employee, satisfied employee index, and marketing expenses/customer
Visualization	Skandia AFS indicators are grouped into the 'FLINK' index – it is possible to analyze performance graphically (pie-chart, trend, area) and statistically, as shown in the first annual report (Skandia, 1994).

2.2.7 Integrated Performance Measurement System

The Integrated Performance Measurement System (IPMS) is an innovative approach for measuring and improving performance specifically designed for smaller organizations and used by 93 of them in Finland (Laitinen, 2002).

The performance viewpoint, as shown in Figure 2.21, is divided between external performance factors (financial and competitiveness) and internal performance factors (costs, production factors, activities, products, and revenues), with which more questions and measures are associated.

According to Laitinen, most organizations stress the importance of company-level profitability, as opposed to manufacturing companies which are more interested in customer-oriented indicators.

There is no implementation guideline available, but the model can be used in any type of organization to manage and improve performance. The importance of each factor will vary from one organization to another, but the applicability of the model will not be affected.

Table 2.18 Summary – IPMS model

Model Name	Integrated Performance Measurement System (IPMS)
Reference	(Laitinen, 2002; Laitinen, 2009; Laitinen and Chong, 2006) (Laitinen, 2002) Google : Citation Count- 80 (10.11.2010) and Scopus Citation Count-31 (16.11.2010)
Origin	Finland
Purpose	Measuring and improving performance, as designed especially for small organizations, is "based on relevant factors and their related dimensions with a view to creating an integrated system" (Laitinen, 2002) based on a managerial view.
Usage	• Used by small organizations in Finland. • Laitinen and Chong use the concept to analyze performance in small organizations UK (10) and Finland (27): here the organization is restricted to fewer than 20 employees. • IPMS is used to analyze managerial jobs and the importance of job-relevant performance information (Laitinen, 2009).
Viewpoints	No. of viewpoints: 3
	• Internal Performance Factors • External Performance Factors • Environmental and non human production factor performance (Laitinen, 2002; Laitinen and Chong, 2006)
Indicators	The indicators, relations between indicators, the object (one main question is addressed), and measures are shown in Figure 2.21. • **Costs** – Performance (costs) = "efficiency in the allocation of costs between production factors" (Laitinen, 2002). Benchmark measures and actual measures are defined for this indicator. • **Production** – Performance (production) is a function of the product's capacity utilization and its performance level in terms of the organization's readiness to operate efficiently. • **Activities –** performance is defined as a function of: time, cost, and quality. **Products –** are defined as a function of: Quality, Flexibility, and Innovativeness. "The ability of the products to satisfy the standard needs of customers is here referred to as quality, the ability to satisfy special needs as flexibility, and the ability to satisfy future needs as innovativeness" (Laitinen, 2002). The measures related to innovativeness.

Table 2.18 Summary – IPMS model (continued)

Indicators	• are related to the effort expended on innovation and the results of that effort. • **Revenue (profitability)** – several measures developed based on customer profitability and product profitability. • **Competitiveness –** performance is defined as a function of growth and market share. This indicator will help provide additional information for the financial results. • **Financial** – performance defined as a function of: profitability, liquidity, and capital structure. "Profitability is regarded as the most important dimension…since it affects liquidity…and capital structure… Liquidity and capital structure measure the firm's short- and long-term solvency" (Laitinen, 2002). **Environmental and non human production –** important considerations for the consumer in modern society (Laitinen, 2002; Laitinen and Chong, 2006)
Visualization	No visualization approaches or techniques are suggested.

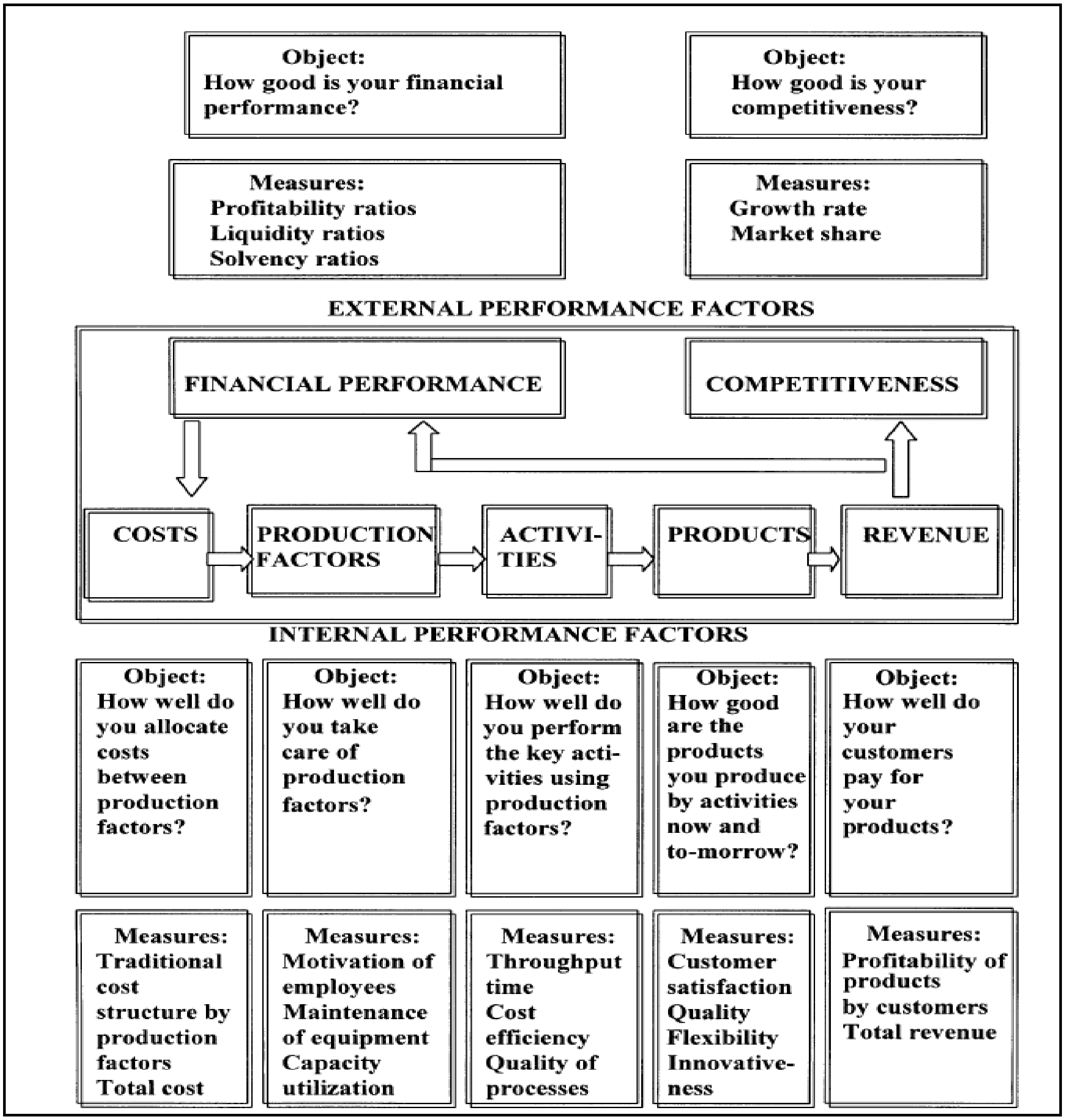


Figure 2.21 IPMS with examples of measures
Taken fom Laitinen (2002, p.79)
(Reprinted from the Scandinavian Journal of Management, vol.18, Laitinen, Erkki K.,
A dynamic performance measurement system: evidence from small Finnish technology companies, pp. 65-99,
Copyright (2002),with permission from Elsevier.)

2.2.8 Sink and Tuttle model

The performance management process must manage what gets done and how these things get done (Sink and Tuttle, 1989, p. 34). Sink and Tuttle claim that the organizations has the responsibility of getting the job done on time, within quality specifications, and with the right amount of resources, and to continuously improve individual, group, organizational, and performance systems (Sink and Tuttle, 1989, p. 34).

The performance management process is an open framework with a feedback loop. Measurement is separate from evaluation. The process is driven towards control and the accomplishment of longer-term goals. As shown in Figure 2.22, a planning process used to develop performance improvement has at least eight areas.

The model claims that the performance of an organizational system is a complex interrelationship of seven criteria (Sink, 1985; Sink and Tuttle, 1989), as shown in Figure 2.23, where profitability is impacted by all the other criteria. According to these authors, innovation and work life quality will influence the input to productivity, which is highly correlated to profitability. The operational definitions of all criteria are presented in Table 2.19 and in section 1.13. In their opinion, the definition of quality is changing from conformance to the designer’s specifications to making sure that the product specifications meet the customer’s specifications.

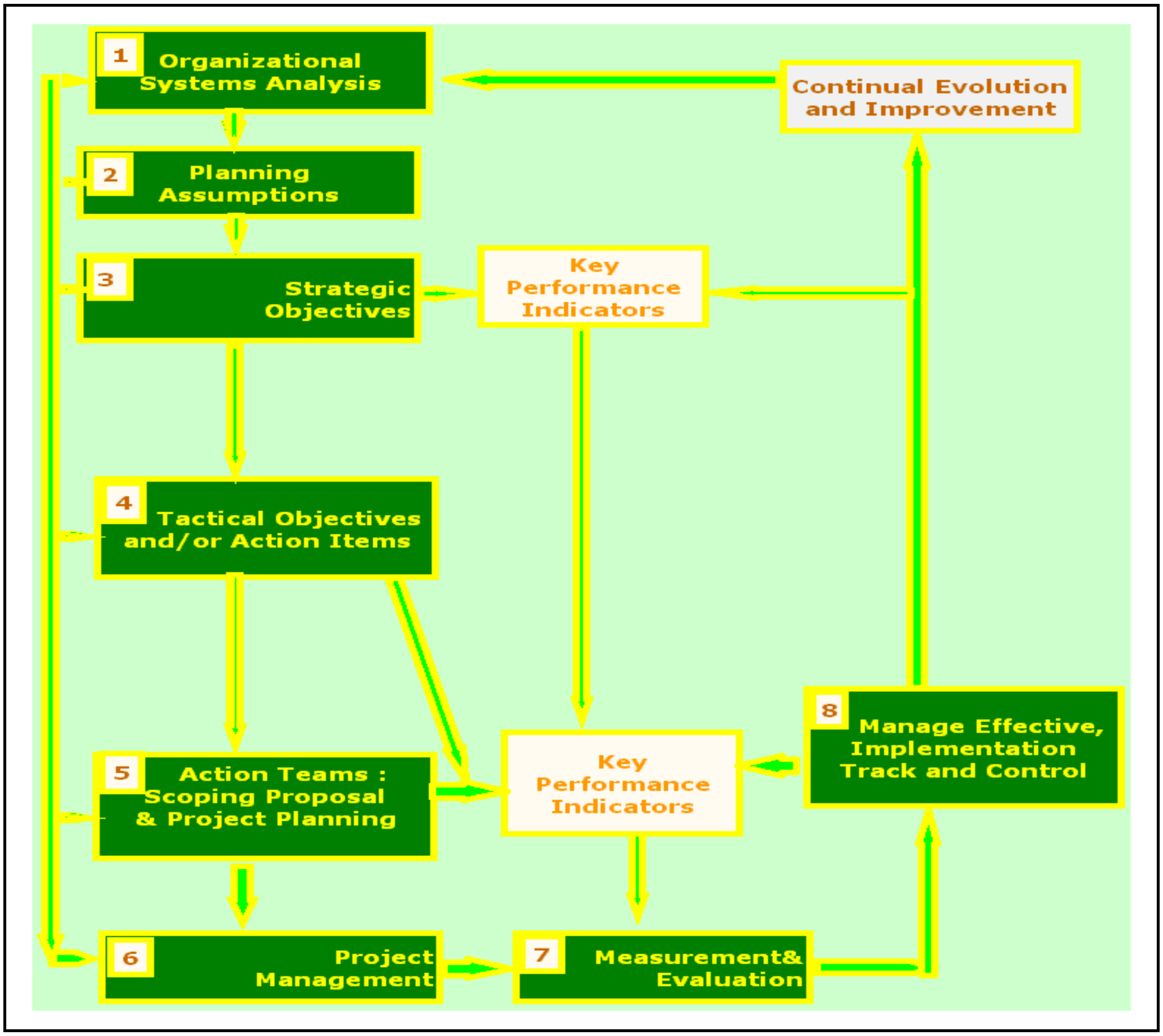


Figure 2.22 Performance improvement planning process.
Adapted from Sink and Tuttle (1989, p.40

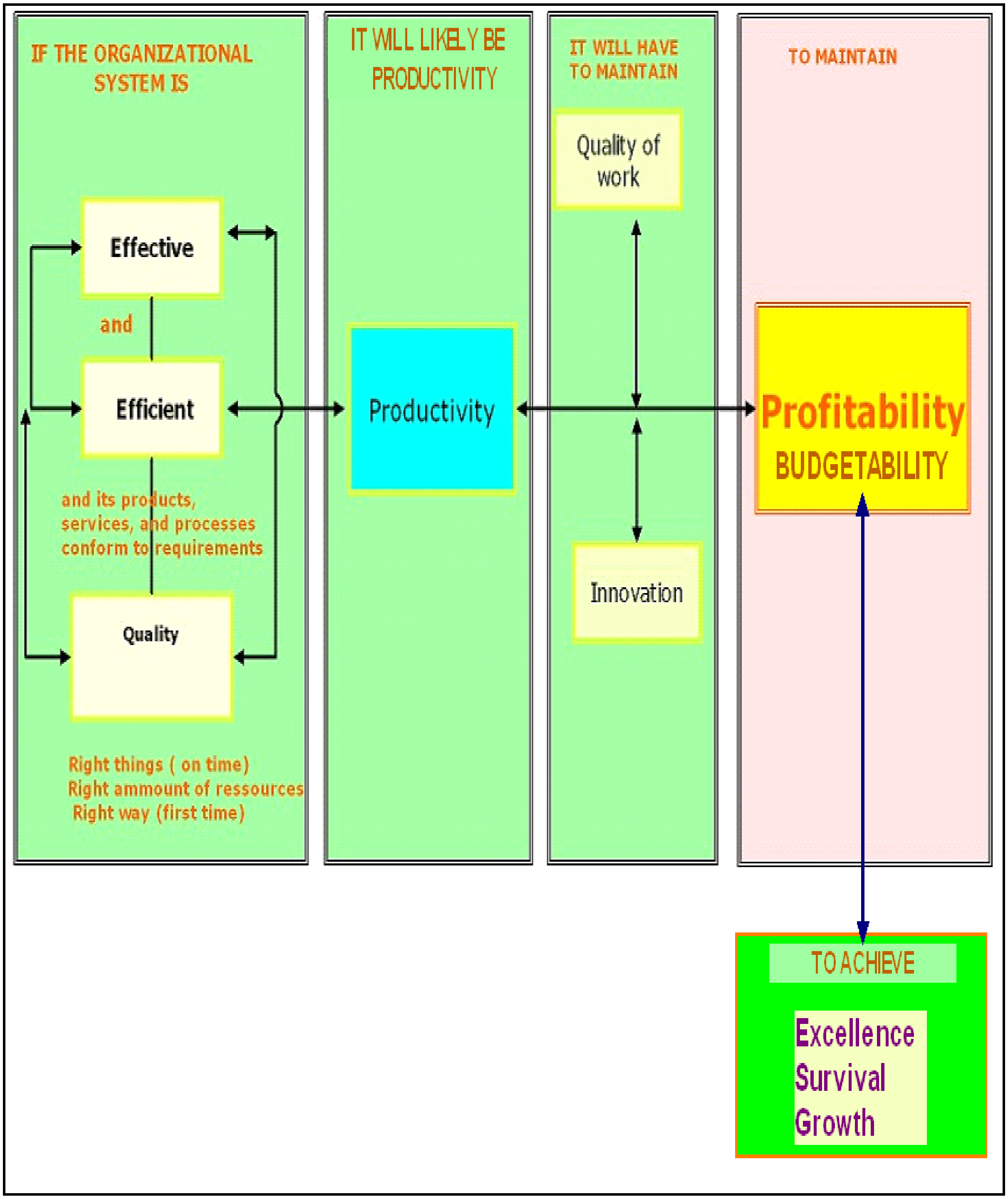


Figure 2.23 Sink and Tuttle model – Interrelationships between viewpoints. Adapted from Sink and Tuttle (1989, p.187)

Table 2.19 Summary – Sink and Tuttle model

Model Name	The Sink and Tuttle Model
Reference	Developed by Sink and Tuttle
Origin	(Sink and Tuttle, 1989) Getcited.org Citation Count- 211 (16.11.2010) (Sink, 1985) Getcited.org Citation Count- 213 (16.11.2010)
Purpose	The purpose is to facilitate communication of the vision of the organization in order to improve performance.
Usage	U.S. Department of Energy (DOE), Energy Information Administration (Education, 2010) . U.S. Government Performance and Results Act of 1990 (GPRA) (OPM, 1996)
Viewpoints	No. of viewpoints: 7
	Effectiveness is expressed as a ratio of actual output to expected output: if the number is larger than one, actual performance is better than target performance; **Efficiency** is defined as a ratio of resources expected to be consumed to resources actually consumed: an organization could be effective but not efficient, or efficient but not effective; **Productivity** is the ratio of output to input; **Quality of work life** – employee attitudes- important factor between productivity and profitability: poor performance in the area of work life quality could provoke failure for an organization; **Innovation** is introduced by the organization in order to sustain and improve performance; **Profitability**, in terms of the profit generated by the organization; **Quality**, where quality is an extremely broad concept. To make the term more tangible, quality is measured at six checkpoints and operationally defined for at least five checkpoints. Sink and Tuttle claim that quality is very important for organizations.
Indicators	Examples of the indicators and how to apply the framework are discussed in (Sink, 1985; Sink and Tuttle, 1989).
Visualization	• None suggested

2.3 Comparison of models

This section presents some observations and a comparison of the models in the previous two sections (see Table 2.20 and Table 2.21). Over the past 30 years, performance management and quality models have become increasingly important to organizations, and many models and approaches have been proposed. As is shown in Table 2.20, presented in the same order as in Section 2.2, a significant portion of the models are from the U.S.: half of these listed in

Table 2.20. Five of the models listed in Table 2.20 originated in Europe, and one is an international standard.

The models analyzed in this literature review can be classified in two categories: software engineering and management. Most of those in Table 2.21 are from the field of management, rather than from the field of software engineering. Table 2.21 also shows that most of the models are closed, meaning that their viewpoints and indicators cannot be easily changed. An important advantage of open models is that viewpoints and indicators can be renamed, added, or deleted. Standardized performance viewpoints and indicators that are decided and designed by head office do not necessarily fully reflect the specific business circumstances of each site or department that wishes to improve its performance.

Table 2.20 Origin of the model – Year and country/region

Name	Year	Country/Region
McCall Model	1997	US
Boehm Model	1978	US
ISO 9126 Model	1991	International standard
Donaldson and Siegel Model	2001	US
Integrative Framework for IS Quality Management	2000	US
QEST & Lime Model	1998	Canada/Italy
Balanced Scorecard (BSC)	1992	US
Performance Prism	1998	UK
EFQM Excellence	1988	Europe
IAM	1997	Sweden
Baldrige Framework	1987	US
Skandia Navigator	1994	Sweden
IPMS	2002	Finland
Sink and Tuttle Model	1998	US

Although consolidated values are important to top managers, Table 2.21 shows (in the same order as presented in Section 2.2) that few models offer any kind of a consolidation technique. Clearly, integrating individual measurements into a single performance value will facilitate good decision making over time, and for this a graphical representation is necessary.

The closest model in the literature to the research problem of this thesis is QEST-nD, an open model which produces a single performance value and takes into account various viewpoints individually and concurrently.

Table 2.21 Comparison of models

Model Name	**Viewpoints**	**Type of model** (Software Engineering (SE) or Management)	**Number of Viewpoints** (at the first level)	**Consolidated Value** (Yes: there is a mathematical formula for consolidation into one global index; No: there is no such mathematical formula)	**Open or Closed** (Open: viewpoints can be added, changed, renamed, or deleted, according to management objectives; Closed: viewpoints cannot be added, changed, renamed, or deleted)
McCall	Revision, Transition, Operation	SE	3	No	Closed
Boehm Model	Portable, Maintainable, As-Is Utility.	SE	3	No	Closed
ISO 9126 Model	External Quality, Internal Quality, and Quality in Use.	SE	3	No	Closed
Donaldson and Siegel Model	Generic	SE	N	Yes	Open
Integrative Framework for IS Quality Management	Administrative, Data, Software, Infrastructure, and Information.	SE	5	No	Closed
QEST-3D	Economic, Social ,Technical	SE	3	Yes	Closed
QEST-nD	As selected by the user	SE	N	Yes	Open
BSC	Financial Customer Internal process Learning	Management	4	No	Closed
BSC Second Generation	Financial Customer Internal process Learning + additional viewpoint(s).	Management	N	No	Open

Table 2.21 Comparison of models (continued)

Model Name	**Viewpoints**	**Type of model** (Software Engineering (SE) or Management)	**Num-ber of View-points** (at the first level)	**Conso-lidated Value** (Yes: there is a mathe-matical formula for consolida-tion into one global index; No: there is no such mathe-matical formula)	**Open or Closed** (Open: viewpoints can be added, changed, renamed, or deleted, according to management objectives; Closed: viewpoints cannot be added, changed, renamed, or deleted)
Performance Prism	Stakeholder Satisfaction, Stakeholder Contribution, Strategies, Processes, Capabilities	Management	5	No	Closed
EFQM	Leadership, Policy and Strategy, People, Partnerships and Resources, Processes, Customer, Society Results	Management	9	No	Closed
IAM	Growth/renewal, Efficiency, and Stability	Management	3	No	Open
Baldrige	Leadership, Strategic Planning, Customer and Market Focus, Measurement, Analysis and Knowledge Management, Human Focus, Process, Business Results	Management	7	No	Closed
Skandia Skandia	Process, Financial, Customer, Renewal and development and Human	Management	5	Yes	Closed
IPMS	External, Internal, and Environmental and nonhuman production	Management	3	No	Closed
Sink and Tuttle	Effectiveness, Efficiency, Quality, Productivity, Quality of Work Life, Innovation, Profitability	Management	7	No	Closed

2.4 International Software Benchmarking Standards Group data repository

The International Software Benchmarking Standards Group (ISBSG), a not-for-profit organization established in 1994, has set up and maintains a database of software project data that can be used by software project managers. This repository is available to organizations, for a nominal fee, and any organization can use it for estimation and benchmarking purposes.

The objective of the ISBSG is "to develop the profession of software measurement by establishing a common vocabulary and understanding of terms" (ISBSG, 2007, p. 2). There is a standard questionnaire for collecting data, and the terms and measures used by this initiative are clearly defined.

The latest release of the ISBSG Repository, release 11 (R11), contains 5,052 projects that were developed in 29 countries and submitted by 24 countries (U.S. (31% of all projects), Japan (17% of all projects), and Australia (16% of all projects)). The application types most widely used are from Financial Transactions (32%), Production Systems (14.6%), and Management Information Systems 12.6%. Most of the projects have a client-server architecture (39.9%) or a stand-alone architecture (39.8%). The third-generation language (63.5%) and fourth-generation language (33.1%) are the most frequently represented.

The ISBSG believes its projects are from the best performing part of the industry, notably because one of the criteria for including a project in the Repository excludes organizations that do not measure functional size and also projects for which work effort (in person-hours) is not available.

Some of the published applications in this database are: estimation (Abran *et al.*, 2007; Berlin *et al.*, 2009; Bourque *et al.*, 2007; Buglione and Gencel, 2008; Lokan and Mendes, 2009; Stroian, 1999), and benchmarking (Comstock *et al.*, 2007; Gencel *et al.*, 2009; Hill and Lokan, 2008; Lokan *et al.*, 2001).

In 1999, the first tools using a white-box approach for simulation, estimation, and benchmarking based on the ISBSG repository were built (Abran *et al.*, 2002; Stroian, 1999). Other tools have since been developed to use this database for estimation (ISBSG, 2008a), but none is specifically designed for performance management.

2.5 The QEST prototype

The QEST prototype (Abran *et al.*, 2003a) is a performance management tool that is implemented using the Java language programming.

This prototype has three important phases:

- selection and weighting of the measures;
- data collection;
- geometrical representation.

First, for the selection and weighting phase, all the characteristics and sub-characteristics of the ISO 9126 model are included in the prototype, as shown in Figure 2.24 and Figure 2.25. The list of measures is not open, and so they cannot be renamed, added, or deleted.

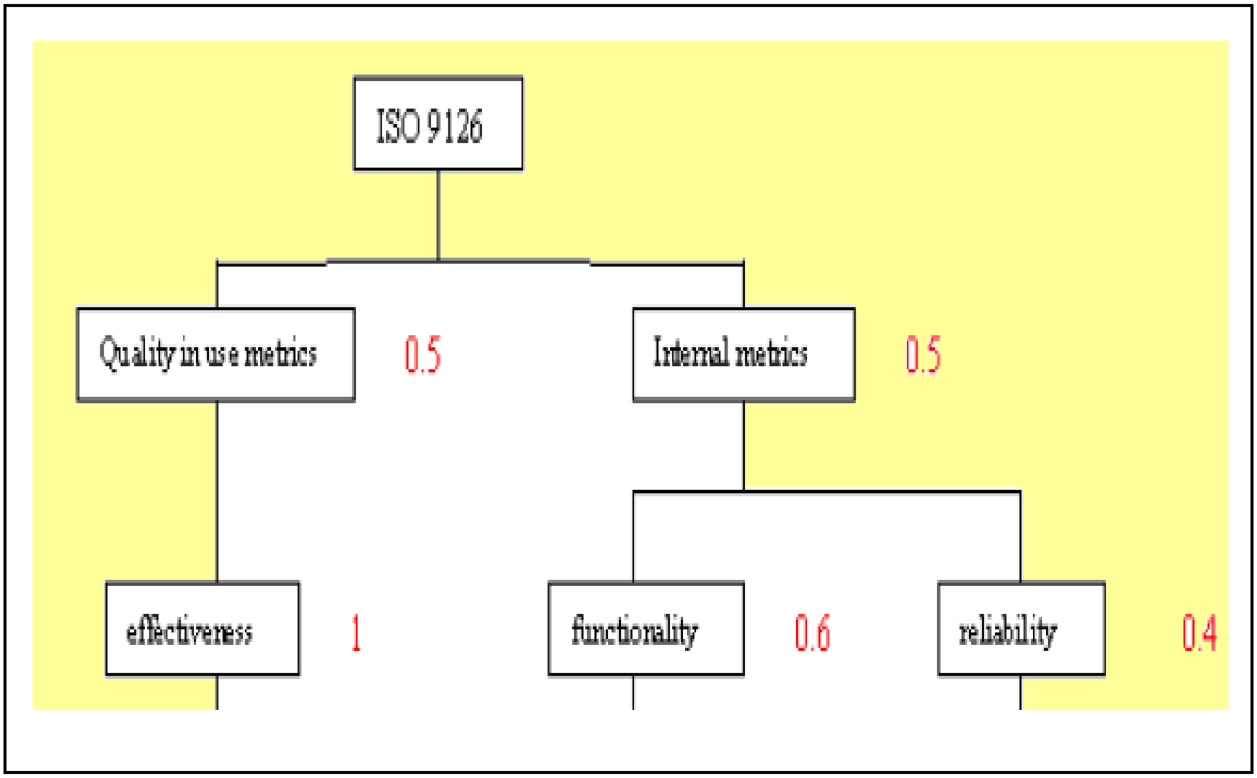


Figure 2.24 Selected nodes.
Adapted from Abran *et al.*(2003a, p. 88)
Reprinted with the permission of the author

The predefined viewpoints in the prototype, as shown in Figure 2.25 and Figure 2.26, are: E (Economic – the managers' viewpoint), S (Social – the users' viewpoint), and T (Technical – the developers' viewpoint). The weights assigned to the individual indicators, as shown in Figure 2.24, are determined by the manager using a scale from zero to one.

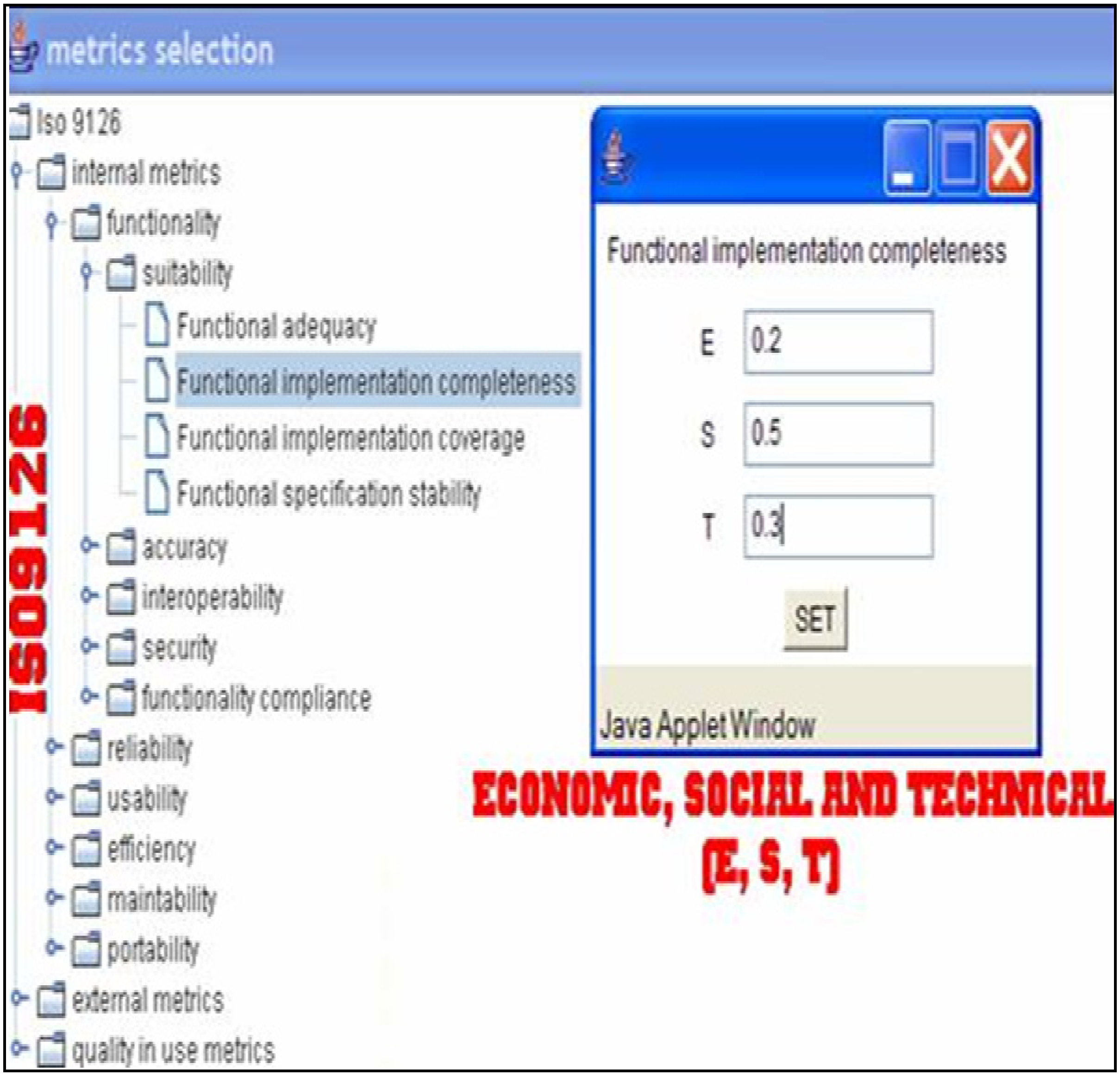


Figure 2.25 Data collection – weighting viewpoint.
Adapted from Abran *et al.*(2003a, p. 87)
Reprinted with the permission of the author

Then, during the data collection phase, the minimum and maximum thresholds values are entered for each measure, as shown in Figure 2.26, so that the prototype can normalize the data on a scale of zero to one.

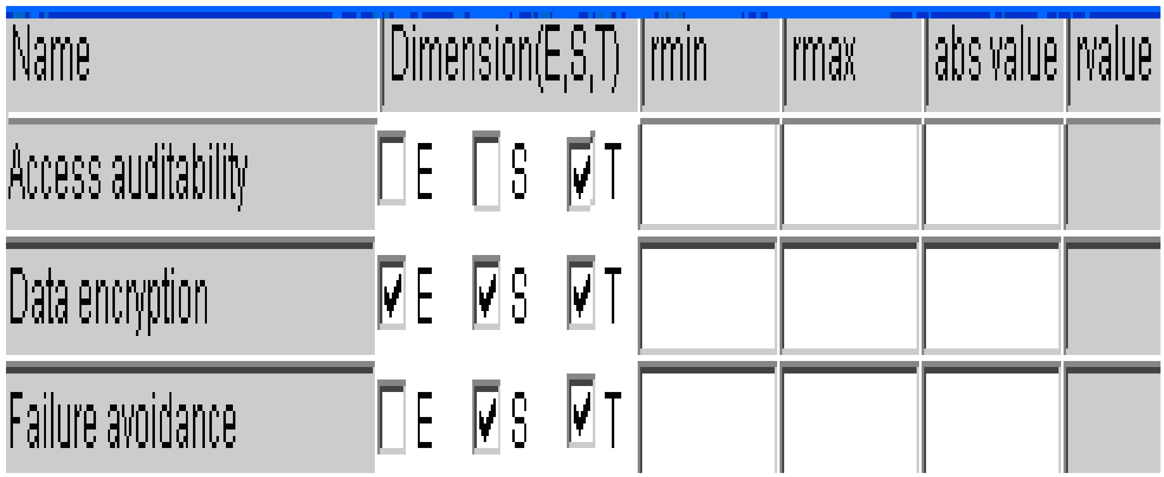


Figure 2.26 Data collection – minimum and maximum.
Adapted from Abran *et al.*(2003a, p. 89)
Reprinted with the permission of the author

To be able to modify data, the manager must restart entering data from the beginning, whether or not this manager has responsibility for that subset of measures. Every measure is well documented and the documentation can be shown on the screen. However, this documentation cannot be modified.

Finally, the geometrical representation is expressed using a tetrahedron. The 3D graphic can be rotated, and the performance values are calculated and normalized for all viewpoints. The volume representing actual performance is colored in green as is shown in Figure 2.27.

The QEST prototype is analysed using the template described in Table 2.22.

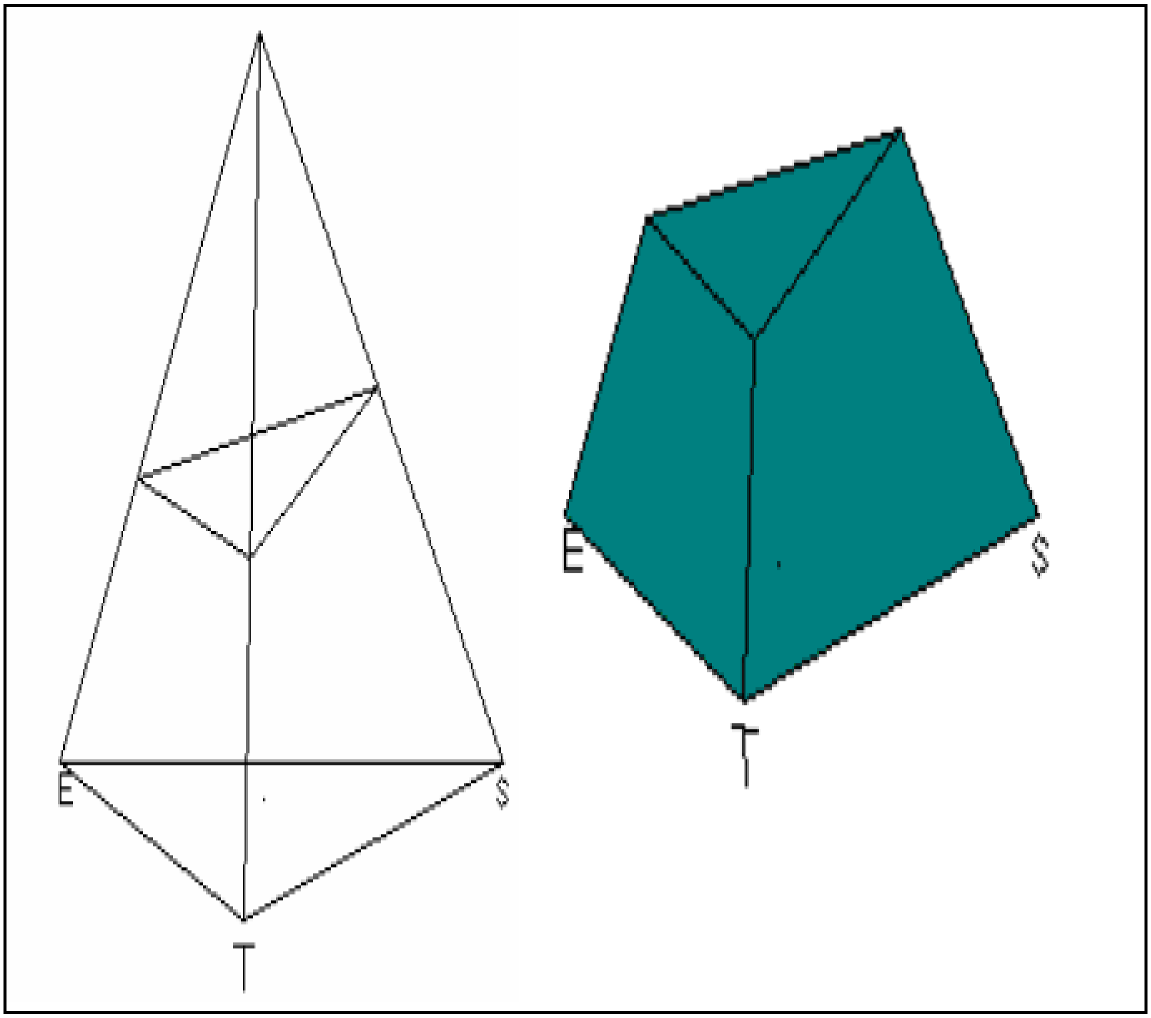


Figure 2.27 Volume visualization
Adapted from Abran *et al.*(2003a, pp. 89-90)
Reprinted with the permission of the author

Table 2.22 Template for identifying and analyzing the QEST prototype

Purpose	What is the purpose of the prototype?
Viewpoints	How many viewpoints does the model support?
Measures	How many measures does the model support?
Data collection	How are performance data collected?
Strengths	What are the strengths of the prototype?
Weaknesses	What are the weaknesses of the prototype?

Table 2.23 Summary – QEST prototype

Purpose	The purpose of the QEST prototype is to manage the performance of software projects through a geometrical representation of performance based on the QEST-3D model and the ISO 9126 standard.
Viewpoints	3 (Economical, Social, and Technical) There is no possibility of choosing more than 3 dimensions, and it is not possible to add, rename, or delete viewpoints.
Measures	27 (ISO characteristics and sub-characteristics) All the measures proposed by the ISO/IEC 9126 standard are included and well documented in the prototype.
Data collection	There is no database or permanent storage mechanism included in the prototype. One of the main disadvantages of this is that the data are temporary. This obliges the user to re-enter the required data at every usage, and current results cannot be compared with historical data or future predicted results. Moreover, there is no minimum and maximum performance value for normalizing measures.
Strengths	• It shows the practical use of the QEST multidimensional model in software quality measurement. • It efficiently uses the 2003 version of ISO 9126, which is the quality model for software products. • It uses a regular tetrahedron to represent performance, which is a simple and visual way to represent performance. • Performance is represented by three distinct concepts: distance, area, and volume. • The ultimate convergence of the three points: Economic, Social, and Technical, to a single point is easy to understand and explain.
Weaknesses	• No visualization techniques are available in the prototype tool for analyzing data and results other than the regular pyramidal representation. Also, visual effects use only one color (green) to represent volume. • It does not include a data repository to store data persistently. • It is limited to ISO 9126 measures and indicators. • The list of measures is not open: they cannot be filled in according to management objectives. • It is limited to the three viewpoints: Economical, Social, and Technical. • Viewpoints cannot be renamed, added, or deleted. There is no possibility of tracking actual performance values against target performance values at different times.

Table 2.23 Summary – QEST prototype (continued)

Weaknesses	• It is not possible to complete benchmarking analyses, because: • there are no historical data; • it is not possible to represent more than one project at a time; • is not possible to store performance target values.

2.6 Discussion

Over the past three decades, there has been an increase in the number of quality and performance models available. This chapter shows that these models:

- differ in how performance is managed, but their ultimate aim is always to manage and improve quality and/or performance.
- vary quite considerably in terms of the terminology adopted, the number of viewpoints included in the model, the chosen viewpoints themselves, and the indicators or measurements within each chosen viewpoint, as shown in Table 2.21. Also, the viewpoints are often determined in advance.
- often do not include the mathematical formulae for consolidating the various performance viewpoints and indicators or measurements into a single index.
- share no consensus regarding the characteristics and sub-characteristics included

One important point is that organizations cannot adopt performance management models 'out of the box', because PMFs must necessarily be adapted to meet particular organizational or project goals and objectives.

More specifically, performance models used by software engineering managers:

- must adopt the terminology and concepts that are recognized by managers and executives outside their own software engineering organizations.
- should support as well concepts and terminology that are specific to software engineering.
- should combine various viewpoints (manager, developer, and user) to represent performance more adequately.

- must represent the analytical requirements of software engineering management when various viewpoints must be taken into account concurrently.

There is no perfect solution to managing performance in software engineering. An optimal solution would adapt and combine concepts and approaches from various models and techniques including:

- elements from the ISO 9126 quality standard;
- the geometrical QEST model. This model includes a consolidated value, but there is currently no tool to use this representation of performance for more than three viewpoints. The model permits the representation of performance in a visual way for three viewpoints but is not possible to represent graphically the results for more than three viewpoints.
- the Balanced Scorecard approach;
- the Prism model based on the geometrical representation of a prism;
- elements from Intangible Assets Monitor (IAM);
- the Sink and Tuttle viewpoints of industrial performance management. These viewpoints could be adapted to the context of software engineering;
- a historical internal/external (ISBSG, etc.) database. PMFs need data to establish targets, to analyze, and to benchmark, and these data are notably missing in many IT organizations. There is currently no performance framework tool that integrates the ISBSG Repository in the context of a PMF for software engineering.

CHAPTER 3

BUILDING A CONCEPTUAL PERFORMANCE MANAGEMENT FRAMEWORK

In chapter 1, the important terms and key concepts associated with Phase 1 and used in this thesis were discussed, and in chapter 2, a review of the multidimensional PFMs found in the literature was presented, the ISBSG Repository was reviewed, and the strengths and weaknesses of the QEST prototype were analyzed (see Figure 3.1).

In this chapter and the next chapter, a prototype for multidimensional performance management in software engineering will be presented, which can represent, graphically and in a consolidated manner, the many possible performance viewpoints, while at the same time keeping track of the values of the individual dimensions.

To build such a prototype, two phases will be completed: build a conceptual PMF, and then develop a software prototype to fully or partially support this framework.

The purpose of this chapter is to build and describe a multidimensional PMF, as shown in the Phase 2 column in Figure 3.1. A prototype implementing this framework will be described in chapter 4.

As shown in the literature review leading up to this chapter, PMFs have been developed to encourage a more balanced view of internal and external viewpoints, and of financial and non financial measures. Performance management is viewed quite differently from one framework to another, and so each framework proposed in the literature adopts its own way of approaching it.

According to Bourne et al., the development of PMFs can be divided into three main phases: the design of the framework, the implementation of the measures, and the use of the measures (Bourne *et al.*, 2000).

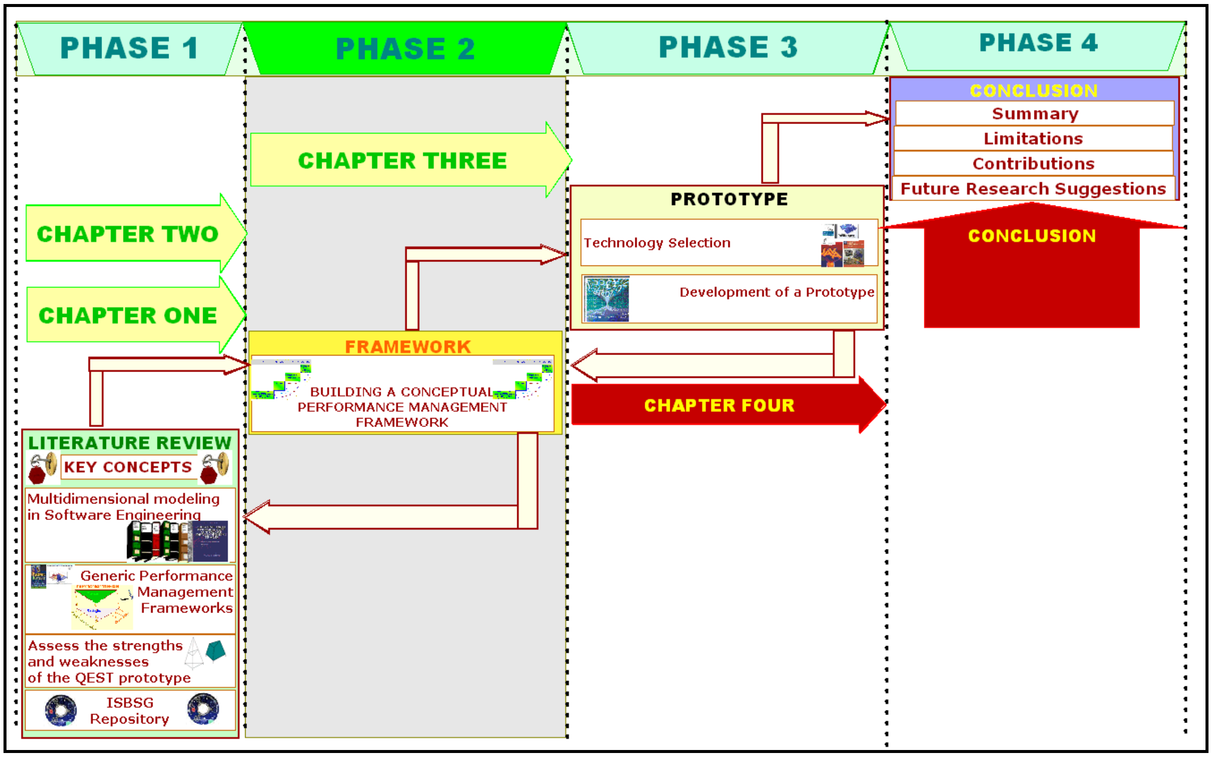


Figure 3.1 Thesis – chapter 3.

According to Abran and Jacquet (Abran and Jacquet, 1997), four phases are required to design a measurement process: design a measurement method, apply the measurement method rules, analyze the results, and use the results .

According to the ISO/IEC 15939 standard (ISO/IEC, 2002, p. 15), four phases are required to design a measurement process: integrate procedures, collect data, analyze data, and develop information products and communicate results. The purpose of this standard is to identify "the activities and tasks that are necessary to successfully identify, define, select, apply, and improve measurement within an overall project or organizational measurement structure" (ISO/IEC, 2002, p. 1), and this standard "does not assume or prescribe an organizational model for measurement" (ISO/IEC, 2002, p. 2).

Therefore, the framework proposed in this thesis, which is designed to manage multidimensional performance in software engineering management, as shown in Figure 3.2, is divided into four steps:

- Step 1: design of the framework;
- Step 2: implementation of the measures;
- Step 3: use of the framework;
- Step 4: performance improvement.

Step 1. The design of the PMF is the starting point. The process of deciding what and how to measure performance requires careful consideration. This design phase is important, and, if not done well, then the entire process will fail from the outset. Integration of the measures and creating the linkages between them is very important, and is not an easy task. Selection of the appropriate measures is also critical to this phase.

Step 2. The implementation of measures can be viewed as a data collection process, followed by preparation of these data for analysis. The design and implementation of a data collection toolset may also be required to collect the measures decided on earlier and defined during the design phase.

Step 3. The use of the framework involves using analysis techniques to transform data into information. The manner in which indicators are understood and interpreted can have an impact on the organization. Visualization tools will facilitate interpretation and decision making during this phase, as well as leading to appropriate changes regarding targets and measures. Two classical procedures are used to set up targets: a top-down procedure and a bottom-up procedure. Analyzing performance does not mean only tracking and controlling actual performance values (APV) against target performance values (TPV) individually. Consolidation techniques, when various performance viewpoints must be taken into account, both individually and in a consolidated manner, are also necessary. Appropriately interpreting and understanding the results is obviously very important as well. Visualization approaches and techniques for multidimensional data are being applied in many and varied disciplines, and are seen as a key enablers of organizational performance management in software engineering (Buglione and Abran, 1999; Stroian *et al.*, 2006).

Step 4. Performance improvement occurs when strategic assumptions are challenged (Sink and Tuttle, 1989). An effective PMF will provide proper feedback and facilitate benchmarking against best internal and external practices. In other words, benchmarking is an integral part of performance improvement, and must be oriented towards concrete action and organizational change.

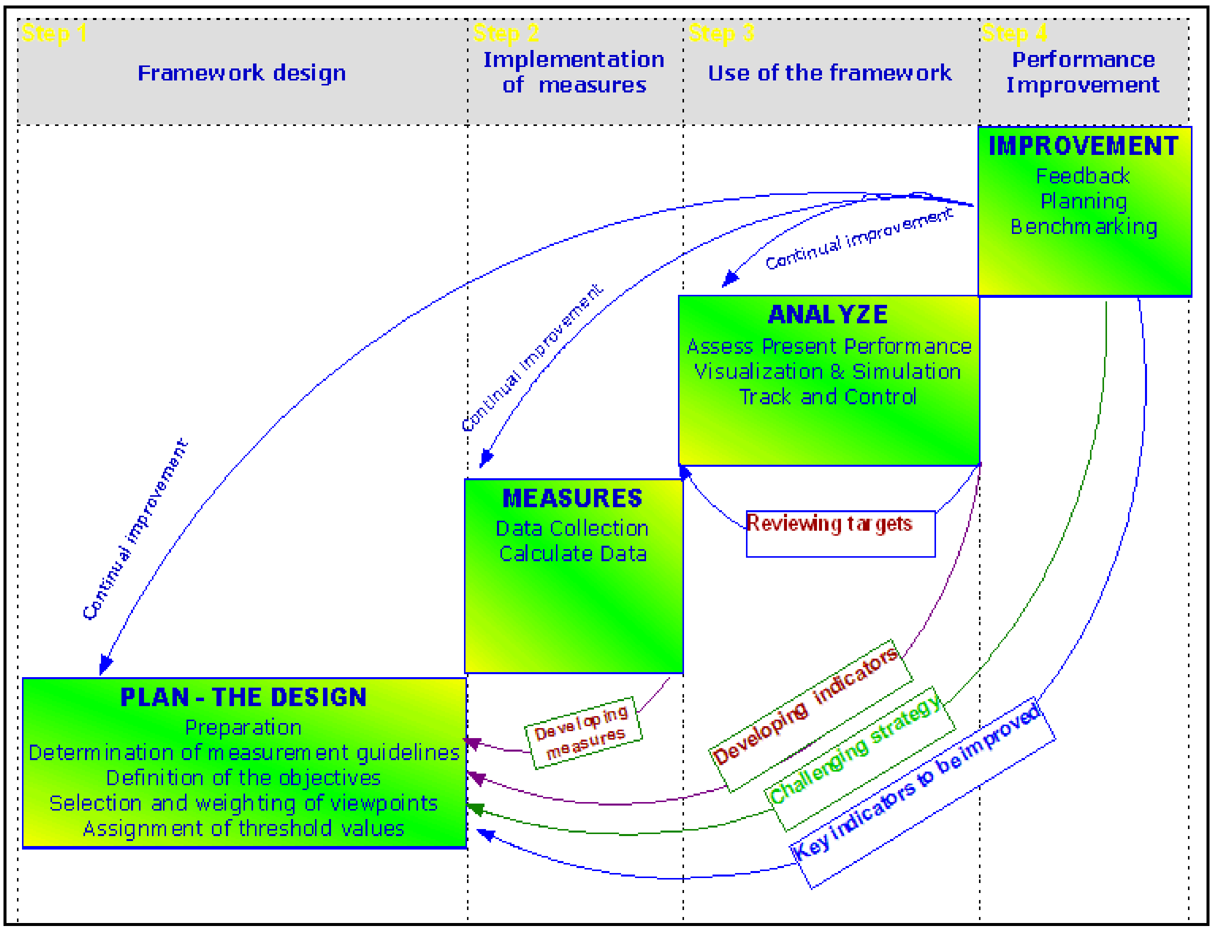


Figure 3.2 Steps in the design of a PMF.

The design of the PMF is presented in section 3.1.1. Then, the implementation of measures is defined and procedures are put in place to collect and calculate the measures as presented in section 3.1.2. The use of the framework is presented in section 3.1.3, and, finally, performance improvement is described in section 3.1.4.

3.1 Designing a conceptual framework to manage multidimensional performance in software engineering

3.1.1 The design of a PMF

As shown in Figure 3.3 the design of a PMF can be subdivided into stages, preparation, determination of guiding elements, definition of objectives, selection and weighting of viewpoints, and assignment of threshold values. According to Flamholtz, a successful performance management system depends on the design step, and, "in the absence of well-designed performance management systems, even the best transformational plan will not succeed because people will not be motivated to support it" (Flamholtz and Randle, 1998, p. 243).

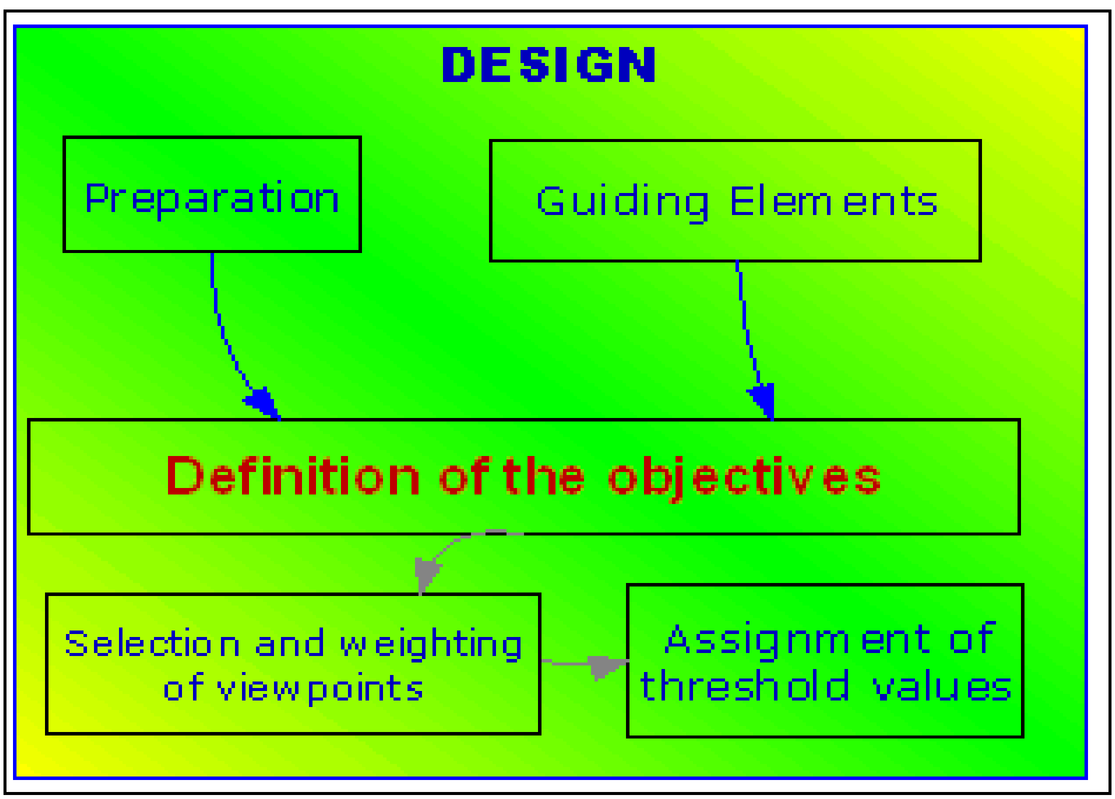


Figure 3.3 Design of a PMF.

3.1.1.1 Preparation

Preparing a management team to design and use a PMF is a critical task, because performance management is part of a complex environment. Preparation has two important steps (Sink and Tuttle, 1989, p. 254): form the team and create a climate in order to support measurement. Every organization has its own distinct performance management culture. Understanding and promoting performance is a challenge and plays a major role in the success or failure of performance management, and so creating and improving a culture favorable to improving performance is a necessity.

Selecting the right human resources is an important activity during this step. According to (Mohrman Jr. *et al.*, 1989, p. 30), at least three types of critical human resources should be included when designing a PMF: senior managers, professionals, and framework users.

3.1.1.2 Guiding elements

Performance management teams try to understand the implicit principles that should be guiding the behavior of their organizations. The assumption is that writing these principles down will clarify them and encourage teams to behave in accordance with them (Sink and Tuttle, 1989). Developing guiding elements must be considered carefully (Sink and Tuttle, 1989). Before performance at any level can be managed, the expectations for that performance need to be clearly established and communicated (Rummler and Brache, 1991). Examples of guiding elements that can be found in the literature are presented in Table 3.1.

Table 3.1 Guiding elements

Meas-urement	• It should be directly related to strategy (Kaplan and Norton, 1996a); • It should be developed in a participatory manner (Fay, 1995, p. 280); • What to measure should ultimately be determined by what the customer considers important (Fay, 1995, p. 278); • It should be a shared vision, not an individual one: "a strategy to which each employee willingly and enthusiastically commits" (Fay, 1995, p. 179);

Table 3.1 Guiding elements (continued)

Meas-urement	• It should be expressed in terms familiar to the organization (Donaldson and Siegel, 2001); • Non financial measures need to be adopted, not only financial ones (Abran and Buglione, 2003; Kaplan and Norton, 1996a).
Goals	• They should be clear, specific, and demanding (Houldsworth and Jirasinghe, 2006).
Improve-ment	• It needs an effective mechanism for reviewing and revising targets and standards (Ghalayini and Noble, 1996); • The measures should provide feedback – recognizing and rewarding excellence – and should stimulate continuous improvement (Sink and Tuttle, 1989); • It should provide clear feedback at all levels (Fay, 1995, p. 291), including appropriate consequences for the best and poorest performers (Mohrman Jr. *et al.*, 1989, p. 248).
Credi-bility	• Organizational performance should be simple enough to be widely understandable, credible, and usable within the organization (Holloway *et al.*, 1995); • Information should be accurate – "The more accurately we measure a business segment's performance, the better informed the decision maker will be" (Giraud *et al.*, 2005, p. 128); • Realistic communication is important: don't promise more than the system can deliver (Mohrman Jr. *et al.*, 1989, p. 136); • Top managers should be involved in the communication plan – their presence will enforce the credibility of the PMF (Mohrman Jr. *et al.*, 1989, p. 137); • Indicators should be based on the most relevant and objective information available (Holloway *et al.*, 1995); • The person who carries the message should be an expert in performance management and have influence within the organization (Mohrman Jr. *et al.*, 1989, p. 137).
Re-sources	• The expertise of first-line managers used in the design phase will have an important impact on organizational success (Judson, 1990, p. 57).

3.1.1.3 Definition of the objectives

Defining the objectives is very important for effective performance management. Performance is not a single, one-dimensional concept: it is not enough to meet a specific target in an unconstrained environment. It is a multidimensional concept that must integrate multiple dimensions which are present simultaneously (Buglione and Abran, 1998). Dimension refers in this thesis to a 'viewpoint', and one of the most common and frequently analyzed viewpoints in performance management in the past has always been the economic one. Distinct but related viewpoints of interest must be taken into account simultaneously, each viewpoint representing a distinct dimension of performance. Organizational goals are the objectives of the organization. According to Sink and Tuttle, the literature is not consistent in the use of these terms, and a clear definition and understanding of them across the organization is a necessity (Sink and Tuttle, 1989, p. 82). At the conceptual level, a goal defines the desired level of performance, and to realize a goal it is necessary to have a question that operationalizes that goal. This question can be divided into k sub-questions. A set of goals (from $Goal_1$ to $Goal_m$) correspond to each viewpoint, and, as shown in Figure 3.6 overall performance is composed of many viewpoints, from $Viewpoint_1$ to $Viewpoint_n$.

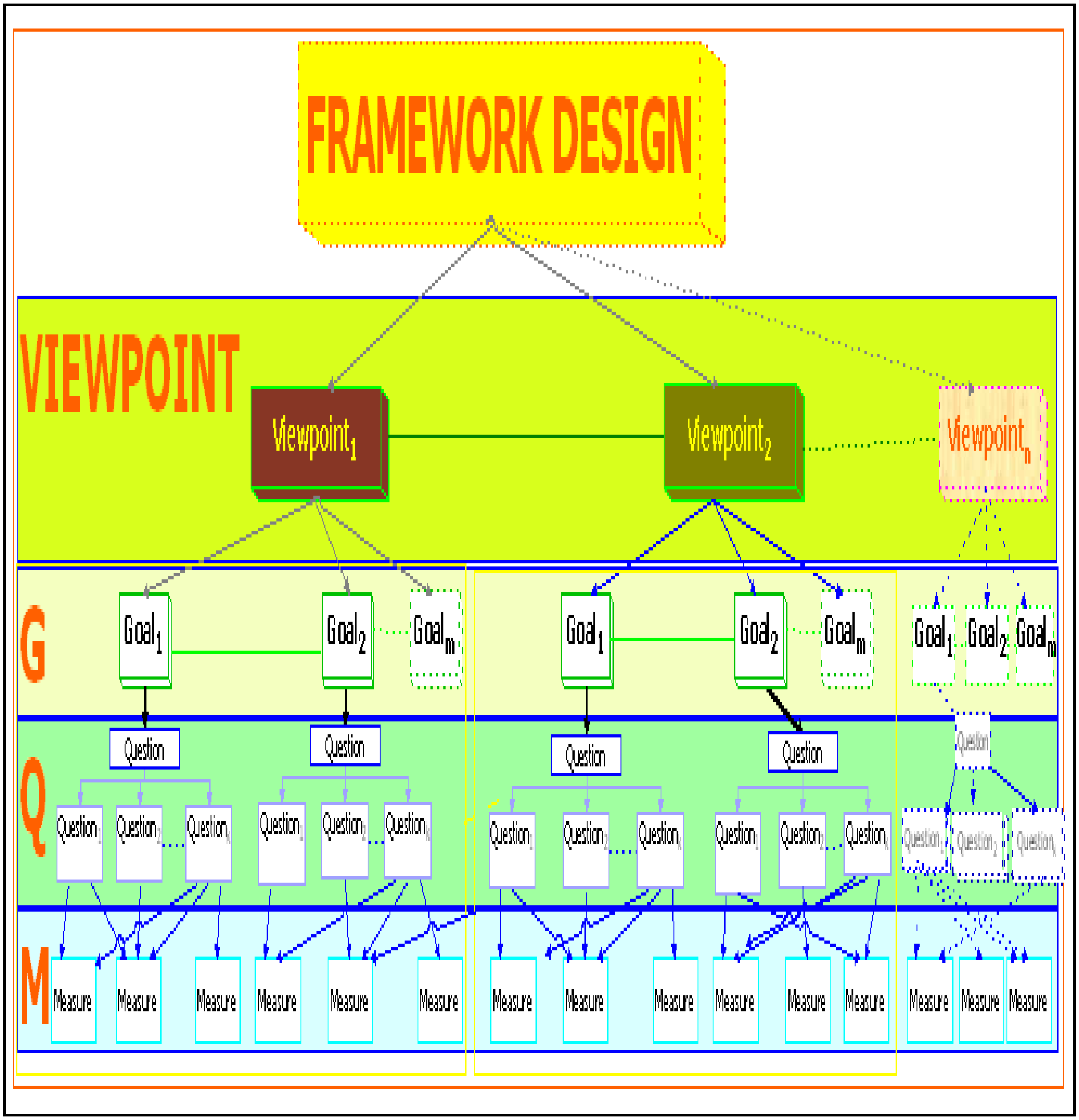


Figure 3.4 Design tree of the framework.

The viewpoints adopted by a selection of multidimensional performance frameworks found in software engineering in particular, and in management more generally, are presented in Table 2.21. The formula for calculating organizational performance as a function of the performance of every viewpoint, as shown in Figure 3.4, is:

$$\text{Performance} = f(\text{Viewpoint}_1, \text{Viewpoint}_2, \ldots, \text{Viewpoint}_n)$$

A set of measures is associated with every question or sub-question in order to have it answered in a quantifiable manner. A set or subsets of related measures form an indicator, which, according to Practical Software Measurement (PSM), is a measure or a group of measures that provides insight into a issue (McGarry *et al.*, 2001). Leading indicators predict future organizational performance, and lagging indicators are measures of previous events. Coincident indicators occur at approximately the same time as the conditions they show.

According to the ISO 15939 standard, and as shown in Figure 3.5:

- an entity is defined as "an object…characterized by measuring its attributes…[that] may have one or more properties that are of interest to meet the information needs…" (ISO/IEC, 2002, p. 21);
- an attribute is a "property or characteristic of an entity that can be distinguished quantitatively or qualitatively by human or automated means" (ISO/IEC, 2002, p. 2);
- a measurement method is defined as "a logical sequence of operations, described generically, used in quantifying an attribute with respect to a specified scale. The operations may involve activities such as counting occurrences or observing the passage of time" (ISO/IEC, 2002, p. 21);
- a base measure is a "measure defined in terms of an attribute and the method for quantifying it" (ISO/IEC, 2002, p. 3);
- a measurement function is defined as "an algorithm or calculation performed to combine two or more base measures. The scale and unit of the derived measure depend on the scales and units of the base measures" (ISO/IEC, 2002, p. 22);
- a derived measure is defined "as a function of two or more values of base measures…[that] capture information about more than one attribute or the same attribute from multiple entities… Normalization of data often involves converting base measures into derived measures" (ISO/IEC, 2002, p. 22);
- an (analysis) model combines measures and criteria using an algorithm;
- an indicator is a "measure that provides an estimate or evaluation of specified attributes derived from a model with respect to defined information needs," and an indicator value is the "numerical or categorical result assigned to an indicator" (ISO/IEC, 2002, p. 3).

"Measurement is always based on imperfect information, so quantifying the uncertainty, accuracy, or importance of indicators is an essential component of presenting the actual indicator value" (ISO/IEC, 2002, p. 22);

- at the interpretation step, quantitative information is related to information needs;
- the information product is defined as "one or more indicators and their associated interpretations that address an information need" (ISO/IEC, 2002, p. 3).

An example of how the various concepts and terms presented above are related is described in the Measurement Information Model (MIM) from ISO 15939 shown on the left in Figure 3.5.

A measure is "a variable to which a value is assigned to represent one or more attributes. The term 'measures' is used to refer collectively to base measures, derived measures, and indicators (McGarry *et al.*, 2001, p. 261). An indicator can be defined as "a measure that provides an estimate or evaluation of specified attributes derived from an analysis model with respect to defined information needs" (McGarry *et al.*, 2001, p. 261). An analysis model involves "two or more base and/or derived measures with associated decision criteria. [It] is based on an understanding of, or assumption about, the expected relationship between the component measures and their behavior over time" (McGarry *et al.*, 2001, p. 23).

Many techniques have been proposed to define and select goals in software development, including the Goal-Question-Metric (GQM), which is widely cited (Basili V.R., 1984; Basili *et al.*, 1994; Park *et al.*, 1996; Rini and Egon, 1999). The Nokia Way (of working) is a loose interpretation of the GQM (Kilpi, 2001). Every organization has a set of goals: for every goal, there is a set of questions, and measures are used to answer the questions.

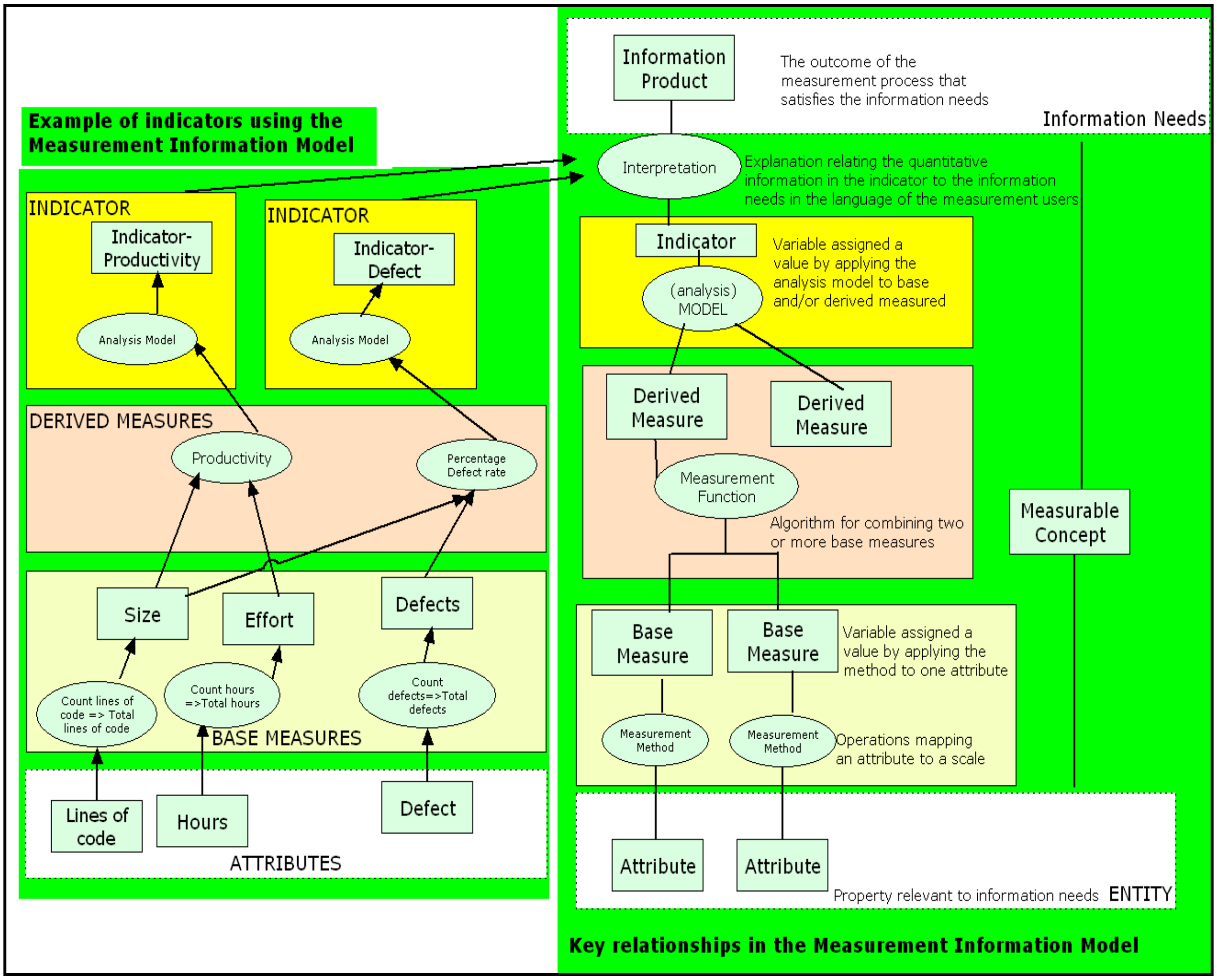


Figure 3.5 Measurement Information Model (MIM).
Adapted from ISO/IEC (2002, p. 20).
(Permission to use extracts from ISO 15939 was provided by the Standards Council of Canada, in cooperation with IHS Canada. No further reproduction is permitted without prior written approval from the Standards Council of Canada.)

The relationship among goals, questions, and measures represented by a GQM tree is shown in Figure 3.6:

1. Identify the goal for the product/process/resource: needs to be challenging, realistically achievable, clear, focused, and easily interpreted.

2. Determine the question(s) that will characterize the way the achievement of each goal is going to be assessed: often more than one question may be necessary to characterize the goal.

3. Define the measure(s) that will provide a quantitative answer – the value of the goal. Every question has one or more measures assigned to it.

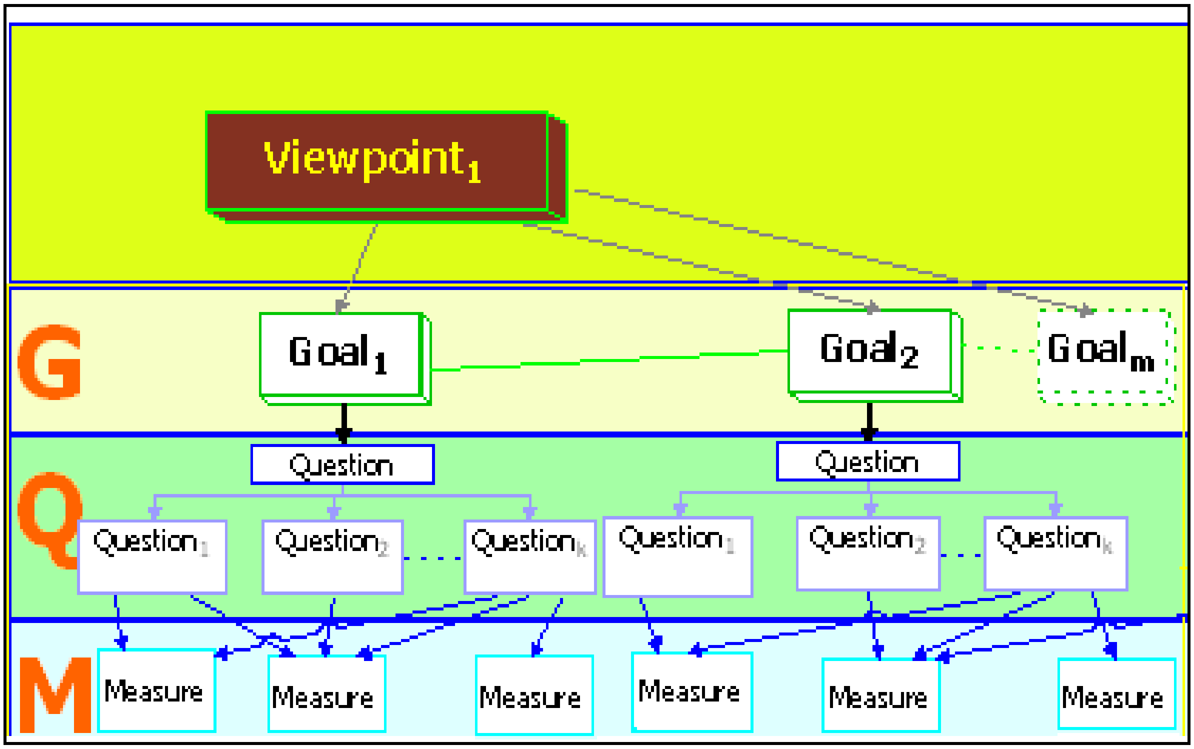


Figure 3.6 GQM Tree.

How are targets set?

Historical performance data help managers improve the setting of performance targets, and setting up targets is an important step for managers. The target is derived from plans, and from the statistical and graphical analysis of historical performance data.

There are two classical procedures to use to set up targets:

- Top-down procedure:
  - A target is fixed for the top level (level 1), and this target will be met only if the target is also met at the level immediately below it (2), and so on until the lowest level is reached.
  - Alignment between strategy and plan is ensured.
  - However, targets may be set at unattainable levels.
- Bottom-up procedure:

- The target is fixed for the lowest level, and the targets for this level determine the targets for the level immediately above it, and so on until the top level is reached.
- It is more difficult to ensure alignment between the strategy and the plan.
- It is easier for managers to motivate the human resources – their participation in setting up the target will generate more buy-in to the targets.

3.1.1.4 Selection and weighting of viewpoints and indicators

A performance viewpoint can be selected for many reasons, depending on strategic or operational considerations. Using inappropriate indicators can have a negative impact on the overall performance of the organization. The ISO 15939 standard gives some examples of criteria for selecting measures and indicators (ISO/IEC, 2002, pp. 13-14; 29-30) .

Organizations that perform very well "set their sights high, and their performance expectations and targets reflect their confidence and optimism" (Fay, 1995, p. 249).

Selection of the appropriate viewpoint – represented by generic names from $Viewpoint_1$ to $Viewpoint_n$ – depends on what decisions they must support: it is important in particular to know who the customer is, what that customer`s goals are, and to select the appropriate measures that will demonstrate whether or not the goal has been achieved.

Few organizations use more than six performance levels (Houldsworth and Jirasinghe, 2006). The Behaviorally Anchored Rating Scales (BARS) approach proposes 7 levels of performance with behavioral ratings ranging from Excellent, the best performance level, to Unacceptable, the worst performance level (Mohrman Jr. *et al.*, 1989, pp. 56-57). An adaptation of this method is the Mixed-Standard Scales approach, where every viewpoint has its own scale consisting of high, medium and low performance (Mohrman Jr. *et al.*, 1989, p. 57).

Three levels of performance are generally very easy to understand: has not met expectations, good all-round or meets all requirements, and exceeds requirements or excellent performance.

Color is considered one of the most significant design tools for communicating. Humans recognize food colors like green or orange, and ‘danger’ color combinations like black and yellow (Kemper *et al.*, 2006).

The symbolic representation developed from the Middle Ages of associating red with fire, yellow with earth, green with water, and blue with air (light and deep colors) has established the four basic colors in the standardized color classification based on instinctive color perception (Kim, 2006). Kim presents a review of the four basic colors expressed in fashion and painting. A fire with a low flame temperature appears in red or yellow, and one with a higher flame temperature is in red, but contains more blue (Kim, 2006).

Using the five levels of performance shown in Figure 3.7 has the advantage of providing a finer level of granularity of differences between levels, and can also make it easier for managers to link performance with rewards. This scale specifies five levels of performance:
1. ‘Unsatisfactory’ is the lowest level of performance, represented by the color red, and is the worst performance in normal operating conditions. It fails to meet requirements.
2. ‘Marginal’ is a less than expected level of performance, but within acceptable limits. The results almost attain the required level of performance, and the performance is still acceptable.
3. ‘Good’ performance meets all expected conditions, and the performance is acceptable.
4. ‘Excellent’ performance goes beyond requirements. This level reflects superior performance results that exceed the expectations of the managers and all activities are achieved or exceed expectations, and the performance is more than acceptable.
5. ‘Outstanding’, the highest level of performance, is usually a challenge for an organization. All activities are not only achieved, but completed in an exemplary manner. This

performance is consistently beyond requirements and greatly exceeds the expectations of managers.

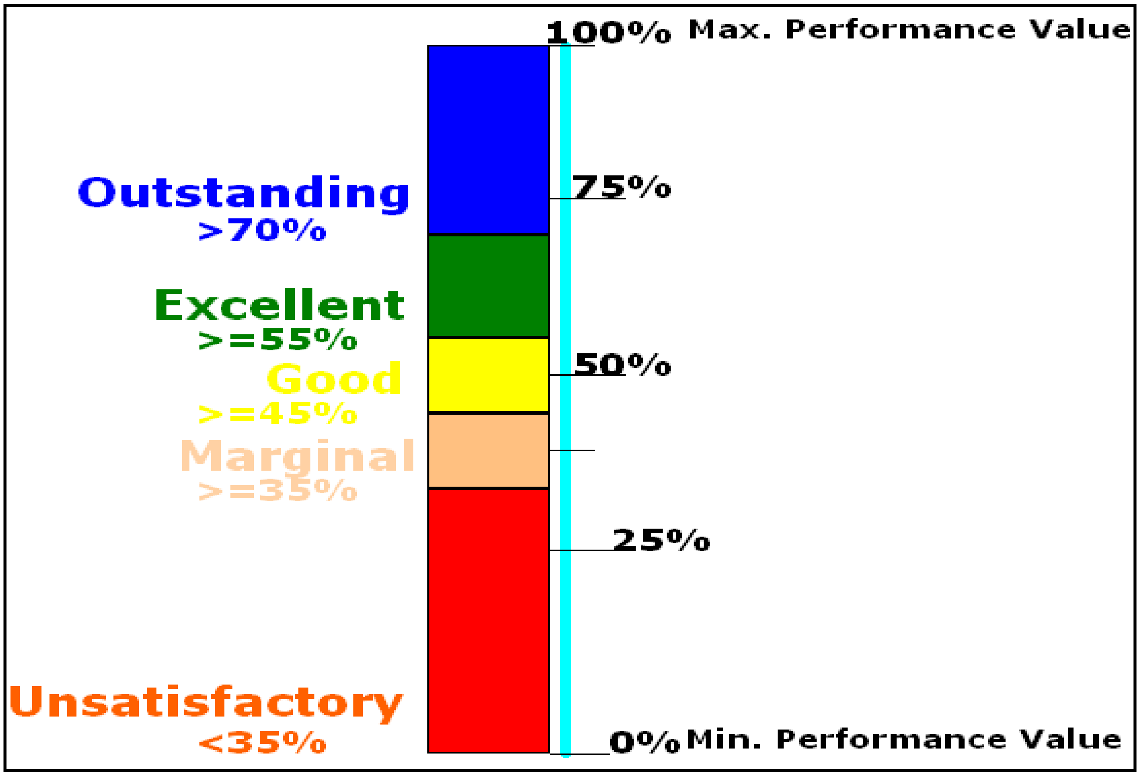


Figure 3.7 Performance scale.

Another way to present these five levels is to group together the two highest levels (Outstanding and Excellent) and call it *superior* level of performance, the two lowest levels (Unsatisfactory and Marginal) and call it an *inferior* level of performance, and, finally, one that is totally acceptable (*good*).

The overall performance is dependent on the weights assigned to the individual indicators. These weightings are determined by the manager, who will determine the importance of the indicators using a percentage scale. Weight assignments provide an opportunity for managers to emphasize certain indicators: if an indicator is critical or most important, it will be weighted most heavily.

During weighting, the manager specifies the importance of each indicator on a scale of 0 (0% = 0 points = minimum) to 1 (100% = 100 points = maximum), as shown in Figure 3.8, and allocates 100 points among the viewpoints:

- The total weight for any indicator must equal 1 (100%) (see Indicator in Figure 3.8): Weight (*Indicator*) = $\sum_{i=1}^{n} Weight(Indicator_i) = 1$, and for Viewpoint: Weight (Viewpoint) $= \sum_{i=1}^{n} Weight(Viewpoint_i) = 1$;
- An indicator can have many sub-indicators, as shown in Figure 3.8: $Indicator_1$, $Indicator_2$, and so on;
- In Figure 3.8, $Indicator_2$ is seen as important, and so it is given a weight of 0.6. $Indicator_1$ is therefore given a weight of 0.4, since the sum of the weights of $Indicator_1$ and $Indicator_2$ must equal 1;
- $Indicator_1$ is then divided into $Indicator_{11}$, with a weight of 0.50, followed by $Indicator_{12}$ at 0.30, $Indicator_{13}$ at 0.10, and, finally, $Indicator_4$ at 0.10. Once again, the sum of these weights must equal 1.

One known method for assigning relative weights is the Analytic Hierarchy Process (APH), which helps managers set the relative importance of indicators or viewpoints (Paulson and Zahir, 1995; Saaty, 1990). Weighting reflects the contribution to the realization of the performance of the organization, as perceived by the managers of each indicator.

The performance for a given viewpoint is highly dependent on the weights assigned to the individual indicators. If the weights assigned to the indicators are not appropriate, then the performance for that viewpoint will be distorted.

The managers can analyze each viewpoint as being of equal importance or of differing importance, as shown in Figure 3.9 .

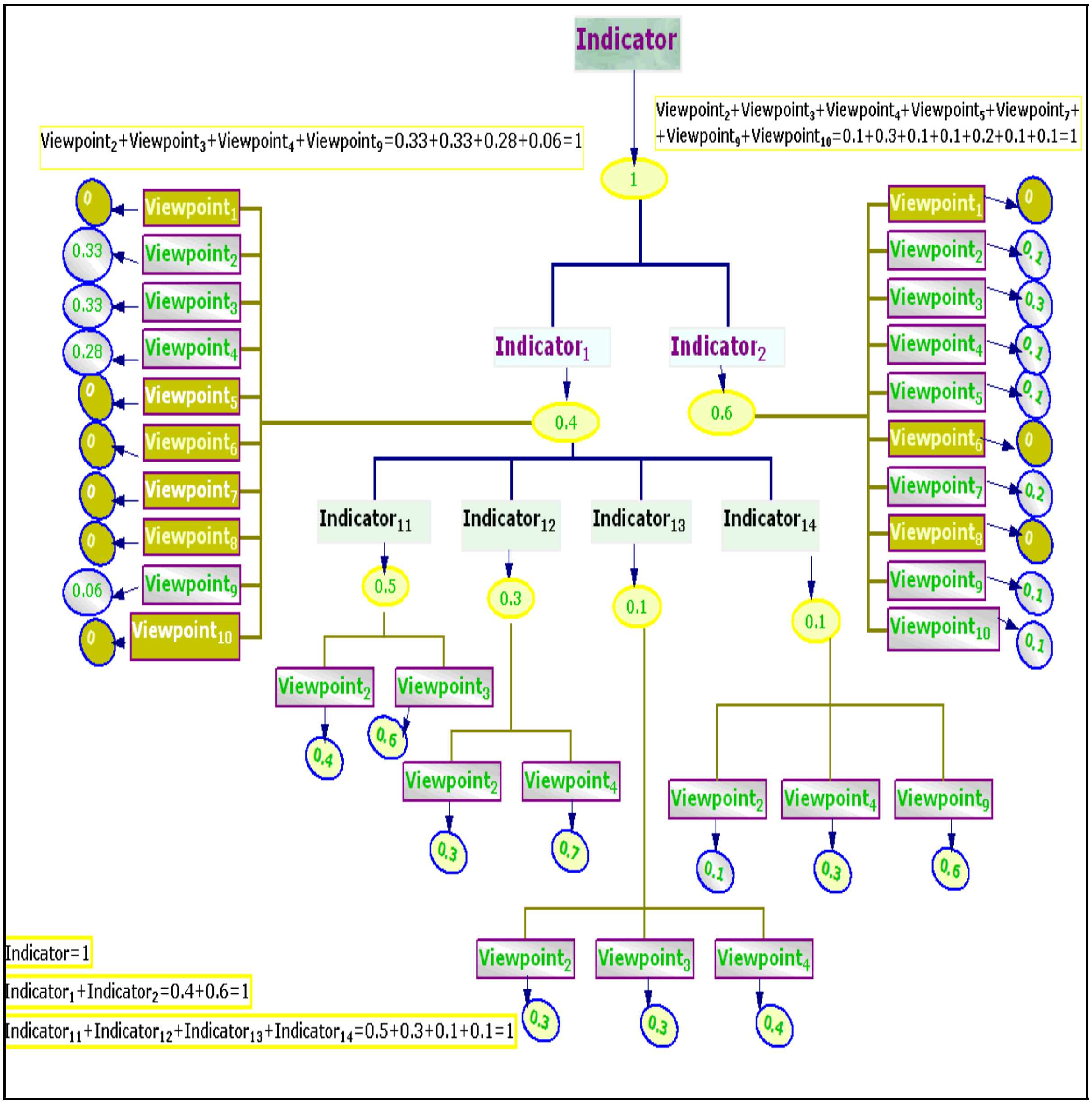
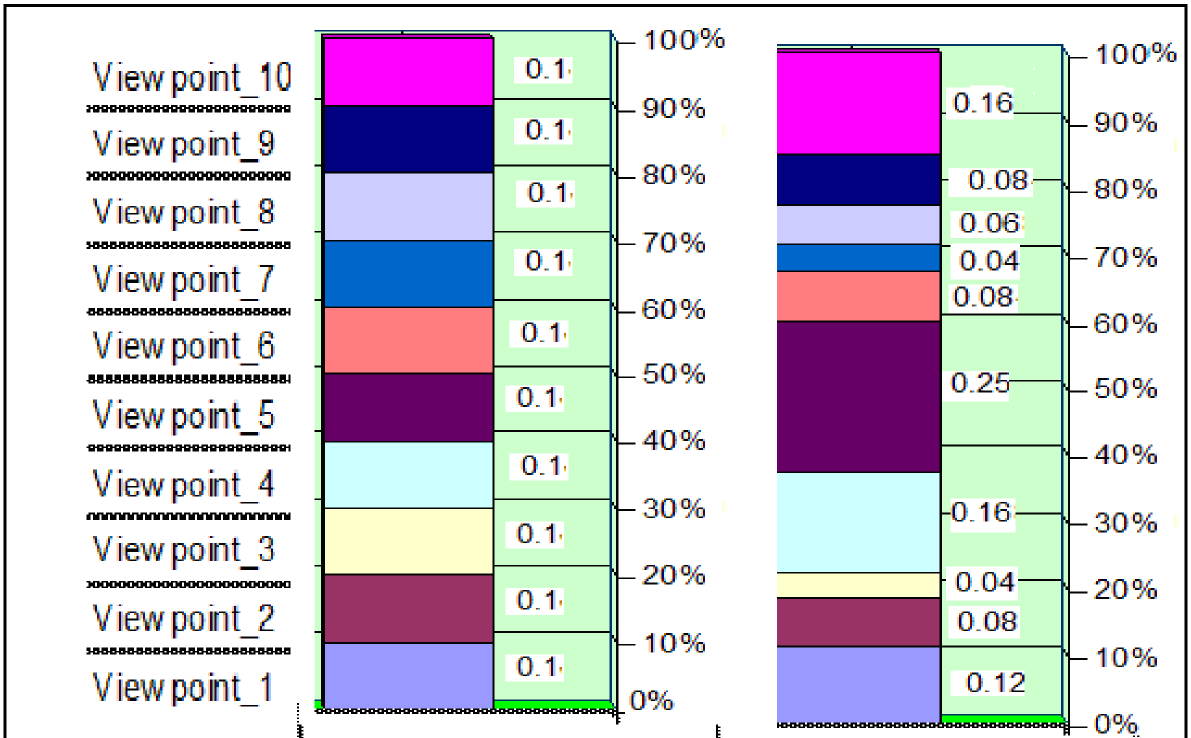


Figure 3.8 Weighting of indicators and viewpoints.



View point_10	0.1	0.16
View point_9	0.1	0.08
View point_8	0.1	0.06
View point_7	0.1	0.04
View point_6	0.1	0.08
View point_5	0.1	0.25
View point_4	0.1	0.16
View point_3	0.1	0.04
View point_2	0.1	0.08
View point_1	0.1	0.12

Figure 3.9 Overall performance: left side-equal weights, right side – different weights.

3.1.1.5 Assignment of threshold values

The target performance value (TPV) is the level at which a performance measure indicates that a goal or a planning objective has been achieved. Assigning the TPV is an important step in performance management.

The threshold values could be different from one viewpoint to another, but they are always between 0% and 100%. In our example, the level of performance in Figure 3.10 is interpreted as follows:

- more than 70% is outstanding performance;
- less than 70% and equal to or more than 55% is excellent performance (acceptable);
- less than 55% and equal to or more than 45% is good performance (acceptable);
- less than 45% and more than or equal to 35% is marginal performance (acceptable);
- less than 35% is unsatisfactory performance (unacceptable).

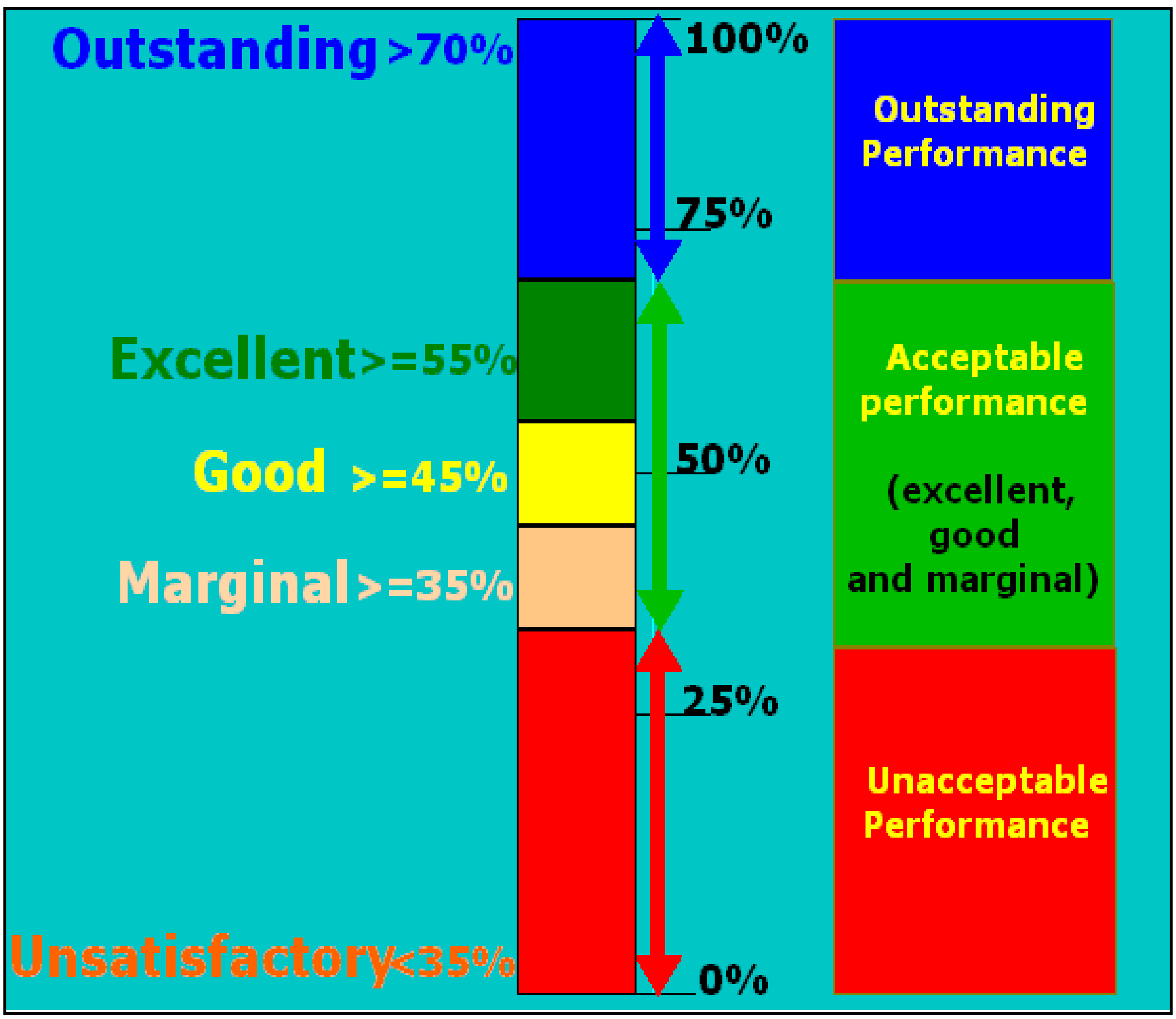


Figure 3.10 The level of performance.

3.1.2 Implementation of measures

The transition between the design and the implementation of measures can be difficult (Bourne *et al.*, 2000). The information is obtained through the data collection phase – data need to be collected and calculated, as shown in Figure 3.11.

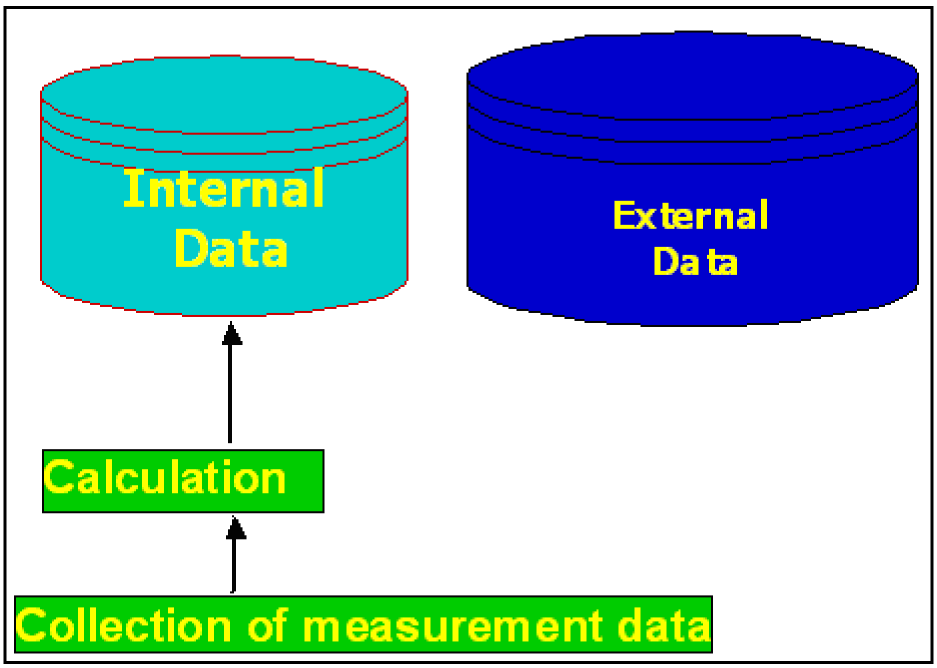


Figure 3.11 Designing – Data collection and calculation.

Collection of measurement data

The quality of an analysis is dependent on the correctness of the source data. If errors exist in the performance data, the analysis technique results will not be of much use: all the analysis may be invalidated. It is also extremely important for the manager to have accurate data in order to make the right decisions. One definition of accuracy in the context of measurement is "closeness of agreement between a measured quantity value and a true quantity value of a measurand" (VIM, 2008, p. 21) and "a measurement is said to be more accurate when it offers a smaller measurement error" (VIM, 2008, p. 21).

Once data requirements have been determined, the sources for the data need to be identified. The use of historical performance data from one's own organization can improve analysis and help managers set up targets:

- There is a need to collect organizational data, such as: target performance value (TPV), actual performance value (APV), Size, Effort, Duration, Defects, and other variables.
- Collecting them as soon as is possible is the best choice for the quality of data.

Another solution for accumulating data is to use external performance data from other organizations working in the same field. Also, the ISBSG (presented in section 2.4) has established and maintains a database of software project data that can be used by software project managers for various purposes, including estimation and benchmarking (ISBSG, 2008b).

Calculation

As shown in Figure 3.5 and Figure 3.12, at the lower level, the base measure value (BMV) is a function of a measurement method (MM) and an attribute (A) (McGarry *et al.*, 2001, p. 25): $BMV_i = MM(A_i)$ where A_i is an attribute (tangible). This is a multilevel hierarchy. The derived measures value (DMV) is a measurement function (MF) of the BMV and/or other DMVs (McGarry *et al.*, 2001, p. 25), as shown in this equation: $DMV_k = MF(BMV_{ik}, DMV_i, \ldots)$.

Indicator

An indicator is created from a set or subsets of related measures. It provides an estimate or an evaluation of specified attributes derived from a model with respect to defined information needs, as shown in Figure 3.5 and Figure 3.12. This concept is very important in performance analysis. Overall performance is dependent on the weights assigned to the individual indicators, as shown in Figure 3.8 and Table 3.2.

The indicator value (IV) (McGarry *et al.*, 2001, p. 25) is a function of an analysis model (AM): $IV_i = AM(BMV_i, DMV_{k,\ldots})$. Every indicator has one or more viewpoints associated with it, as shown in Figure 3.8, and Figure 3.12 and Table 3.2.

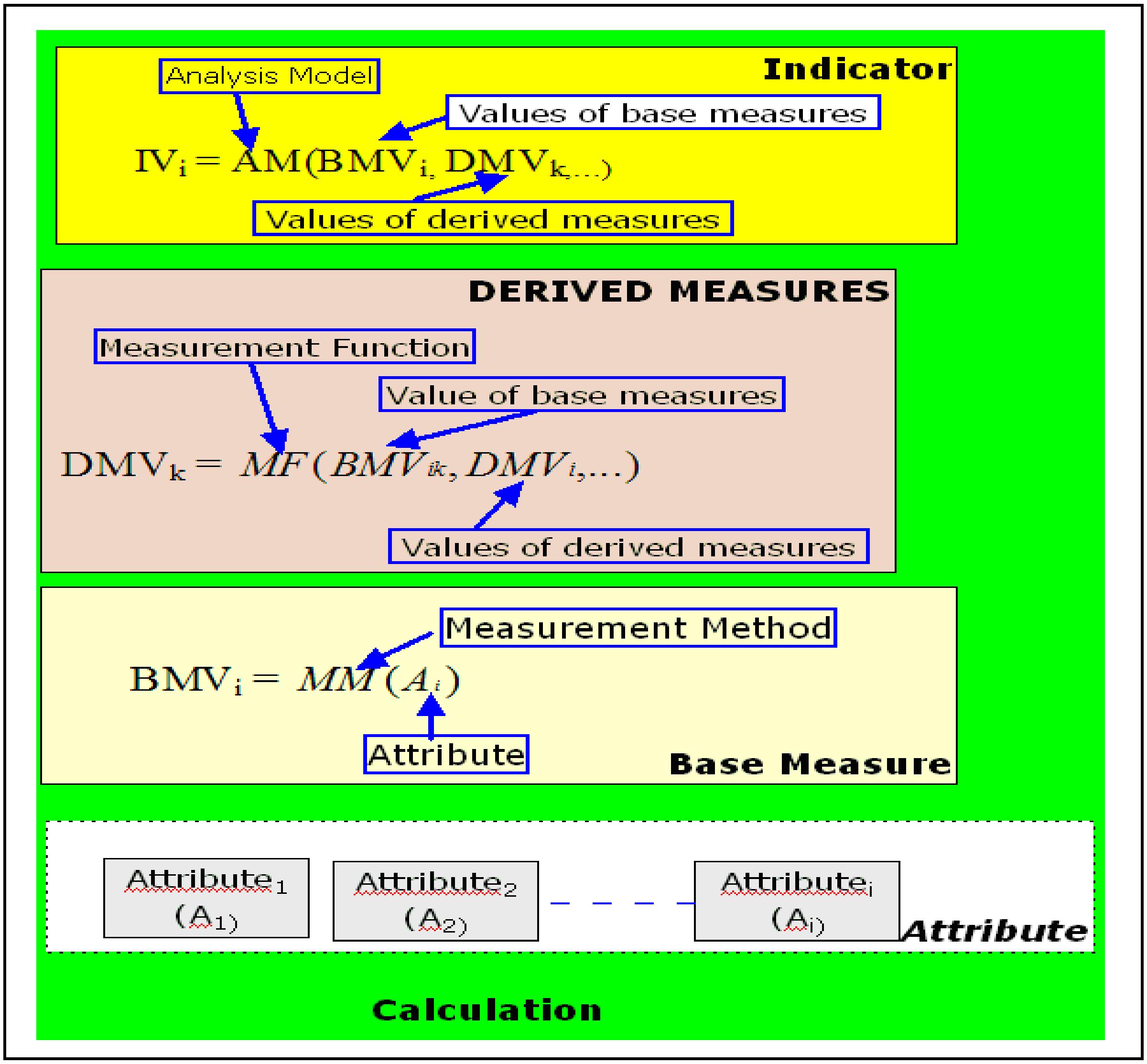


Figure 3.12 Calculation – Theory.

The performance results reflect the overall organizational performance value, and the formula for calculating organizational performance (OP) is a function of the performance of every viewpoint, as shown in Table 3.2 and Table 3.3.

$$OP=f(Viewpoint_1,Viewpoint_2, Viewpoint_3,Viewpoint_4,Viewpoint_5,Viewpoint_6,Viewpoint_7, Viewpoint_8 ,Viewpoint_9, Viewpoint_{10}\ldots Viewpoint_n)$$

Table 3.2 Overall organizational performance

	MEASURE			INDICATOR			PERFORMANCE
Viewpoint	Base Measure Value (BMV)	Derived Measures Value (DMV)	Weight	Indicators	Results Value (IV)	Weight	Performance Value
$Viewpoint_1$	A_1 ---------- A_k	$Measures_o$ ……….. $Measures_p$ -------		$Indicators_m$ …………... $Indicators_n$			f($Indicators_1$, ……………….. $Indicators_n$)
$Viewpoint_2$							
$Viewpoint_3$							
$Viewpoint_4$							
$Viewpoint_5$							
$Viewpoint_6$							
$Viewpoint_7$							
$Viewpoint_8$							
$Viewpoint_9$							
$Viewpoint_{10}$							
……….							
$Viewpoint_n$							
					TOTAL Performance Value OP		APV (Actual (today's) Performance Value) and/or TPV (Target Performance Value)

Table 3.3 Performance table – Summary of TPV vs. APV (from Table 3.2)

Time / Viewpoint	Point- in -time 1 Past	Point- in -time 2 Past	Point-in-time 3 NOW	Point-in-time 4	Point-in-time 5	Point-in-time 6	Point-in-time 7	Point-in-time 8	Point-in-time 9	Point-in-time 10
$Viewpoint_1$	(TPV) / *APV*	(TPV) / (APV)	(TPV) / APV	(TPV)	(TPV)	(TPV)	(TPV)	(TPV)	(TPV)	(TPV)
$Viewpoint_2$	(TPV) / (APV)	(TPV) / (APV)	(TPV) / APV	(TPV)	(TPV)	(TPV)	(TPV)	(TPV)	(TPV)	(TPV)
$Viewpoint_3$	(TPV) / (APV)	(TPV) / (APV)	(TPV) / APV	(TPV)	(TPV)	(TPV)	(TPV)	(TPV)	(TPV)	(TPV)

Table 3.3 Performance table – Summary of TPV vs. APV (continued)

Time / Viewpoint	Point- in -time 1 Past	Point- in -time 2 Past	Point-in-time 3 NOW	Point-in-time 4	Point-in-time 5	Point-in-time 6	Point-in-time 7	Point-in-time 8	Point-in-time 9	Point-in-time 10
Viewpoint$_4$	(TPV) / (APV)	(TPV) / (APV)	(TPV) / APV	(TPV)	(TPV)	(TPV)	(TPV)	(TPV)	(TPV)	(TPV)
Viewpoint$_5$	(TPV) / (APV)	(TPV) / (APV)	(TPV) / APV	(TPV)	(TPV)	(TPV)	(TPV)	(TPV)	(TPV)	(TPV)
Viewpoint$_6$	(TPV) / (APV)	(TPV) / (APV)	(TPV) / APV	(TPV)	(TPV)	(TPV)	(TPV)	(TPV)	(TPV)	(TPV)
Viewpoint$_7$	(TPV) / (APV)	(TPV) / (APV)	(TPV) / APV	(TPV)	(TPV)	(TPV)	(TPV)	(TPV)	(TPV)	(TPV)
Viewpoint$_8$	(TPV) / (APV)	(TPV) / (APV)	(TPV) / APV	(TPV)	(TPV)	(TPV)	(TPV)	(TPV)	(TPV)	(TPV)
Viewpoint$_9$	(TPV) / (APV)	(TPV) / (APV)	(TPV) / APV	(TPV)	(TPV)	(TPV)	(TPV)	(TPV)	(TPV)	(TPV)
Viewpoint$_{10}$	(TPV) / (APV)	(TPV) / (APV)	(TPV) / APV	(TPV)	(TPV)	(TPV)	(TPV)	(TPV)	(TPV)	(TPV)
....										
Viewpoint$_n$	(TPV) / (APV)	(TPV) / (APV)	(TPV) / APV	(TPV)	(TPV)	(TPV)	(TPV)	(TPV)	(TPV)	(TPV)
Overall Organizational Performance	**(TPV) / (APV)**	**(TPV) / (APV)**	**(TPV) / APV**	**(TPV)**	**(TPV)**	**(TPV)**	**(TPV)**	**(TPV)**	**(TPV)**	**(TPV)**

3.1.3 Use of the framework

A PMF should motivate all members of the organization to work in the same, chosen direction. Employees may be motivated if they see performance progressing towards the goals. The benefit of a PMF will be limited, however, if the information is not properly analyzed. In addition, analysis helps to determine whether or not each measure and indicator is effective, and is contributing to reaching organizational goals and objectives.

Analyzing performance does not mean simply monitoring APV against TPV. It is also important, of course, to have a good understanding of why there are deviations in performance from expectations. One of the most common ways to track performance values is to analyze TPV/APV against time, as shown in Table 3.3.

Everyone involved in managing performance must know how the organization is progressing, and it is important to find visualizations that effectively communicate the right information at the right moment.

It is also important to communicate information at multiple levels of detail. For example, managers may see the organization from a functional point of view, notably where goals are established for each function. The optimization of a set of given functions within an organization may indeed contribute to improving the performance of the entire organization, but this process may miss cross functional optimizations. Sink and Tuttle introduced a horizontal view of the organization (Sink and Tuttle, 1989), using the PMF at three levels, as shown in Table 3.4 and Figure 3.14. It is important to link operational indicators and strategic indicators.

Table 3.4 Horizontal view

Level	Range	Scope
Strategic	Long	Performance strategic indicators tell an organization about the soundness of their strategic decisions. They do not replace the strategy, but help put it into practice.
Tactical	Medium	Performance tactical indicators are used for short- to medium-period management. They are used more in a unit of an organization, or for a limited group (a team) or specified project.
Operational	Short	Performance operational indicators help manage operational tasks that are performed on a daily basis.

The performance models that quantitatively represent many measures and viewpoints need to be analyzed and understood. According to Morris, "data…particularly when there is a huge amount of it…need to be organized and processed…[and] must be turned into information]

(Morris, 2003, p. 1). Quantitative methods are “a collection of techniques for organizing, presenting, summarizing, communication, and drawing conclusions from data, so that [they] become informative” (Morris, 2003, pp. 1-2). These methods “need to be interpreted in practical terms before they become useful aids to the solution of business problems” (Morris, 2003, p. 3).

According to (McGarry *et al.*, 2001, p. 85), the three primary types of analysis in software measurement are estimation, feasibility analysis, and performance analysis, as shown in Figure 3.13:

- Estimation is the first type of analysis in any project, and “poor estimates and misconceptions about the estimating process often contribute to failed projects” (McGarry *et al.*, 2001, p. 86).
- Feasibility analysis “evaluates the accuracy and realism of projects plans… To be feasible, the individual elements of the plans must be technically realistic and achievable, and the elements must be consistent in relation to each other” (McGarry *et al.*, 2001, p. 104).
- Performance analysis: assessing the APV against TPV (plans) − lack of planning and unrealistic plans are significant problems.

Analyzing and tracking down a quality (or performance) problem in an organization can be achieved using the seven tools associated with the Japanese quality expert Ishikawa, called Ishikawa’s Seven Tools: check sheets (a frequency table classified in category data), Pareto charts (a bar chart where the items are arranged in order), cause-and-effect diagrams (‘brainstormed’ ideas are structured), stratification (division of data into different categories and sub categories), histograms (showing distributions of items), scatter plots (showing the relations between two variables), and control charts (Morris, 2003, pp. 261-278).

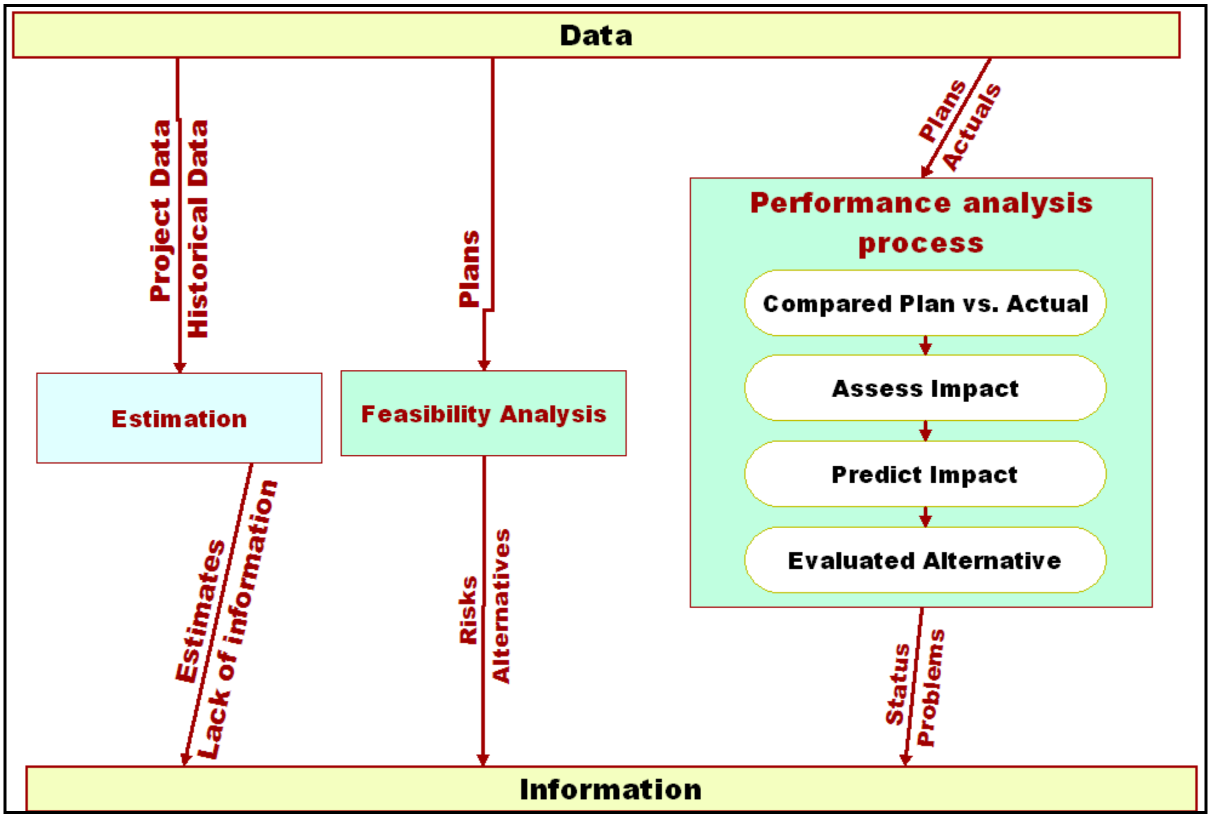


Figure 3.13 Analysis in software measurement.
A synthesis of McGarry (2001, pp.65-124)

According to Dykes *et al* "exploring and analyzing vast volumes of multidimensional data is becoming increasingly difficult" (Dykes *et al.*, 2005, p. 23) without proper visualization tools and approaches. The vast volumes of data "generated by many simulations, the difficulties of managing, exploring, and analyzing simulation data has [*sic: have*] become a real challenge in many disciplines and applications" (Sokolowski and Banks, 2009, p. 104).

Visual analytics is a field of growing interest, because there is "a need to more easily and effectively analyze the mountains of data that organizations are gathering and also to see their findings in ways that are simple to understand and [that] work" (Lawton, 2009) and is focused "on going beyond the interaction with the visual representation of the simulation to the analysis of data" (Sokolowski and Banks, 2009, p. 107). 'Data analysis' was defined in

2005 as "the science of analytical reasoning supported by the interactive visual interface" (Sokolowski and Banks, 2009, p. 107) and incorporates tools "from many fields like knowledge management, statistical analysing, cognitive science, decision science…to maximize [the] human capacity to perceive, understand, and reason…" (Sokolowski and Banks, 2009, p. 107).

According to Dykes *et al* "visual data exploration aims at integrating the human into the data exploration process, applying human perceptual abilities to the analysis of large data sets available…" (Dykes *et al.*, 2005, p. 24). Using visual representation is important "because half the human brain is devoted, directly or indirectly, to vision" (Sokolowski and Banks, 2009, p. 104). Data visualization is used both in the initial exploration before statistical analysis and in the final display of results, as well as in model building.

A selection of one- and two-dimensional or multidimensional graphs for visualizing performance data and setting up performance targets is presented in Table 3.5 and Table 3.6.

Table 3.5 Visualizing performance: One- and two-dimensional graphs

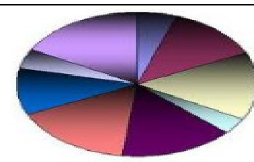
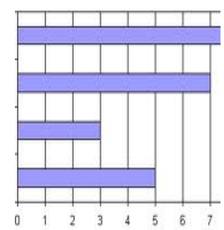
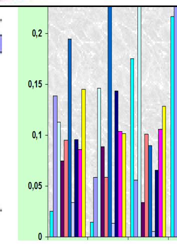
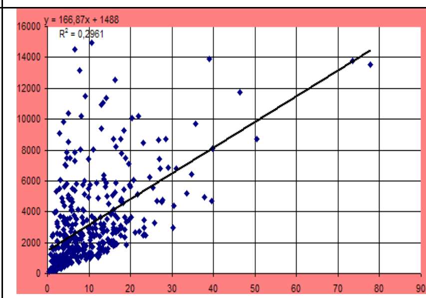
One- and Two–dimensional Graphs	Example
Pie Graphs represent performance data that are broken down into various sectors proportional to the size of the performance represented; they are easy to understand, but limited.	
Bar Graphs (bar charts) can represent the performance of more than one category (e.g. viewpoint, indicator) in the form of bars; this is a comprehensive means of visualizing and tracking performance over time.	 
Regression analysis is used to investigate the relationship among different variables and it "starts with a formulation of the problem. ...determination of the question(s) to be addressed… The problem statement is the first and perhaps the most important step in regression analysis" (Chatterjee and Hadi, 2006, p. 11).	
Bullet Graphs make it easy to highlight a single viewpoint or indicator, and compare APV to TPV.	

Table 3.5 Visualizing performance: One- and two-dimensional graphs (continued)

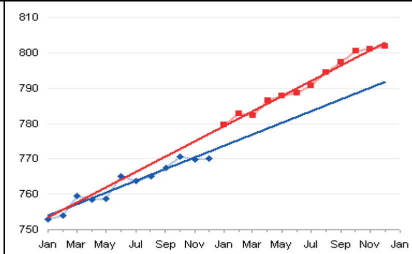
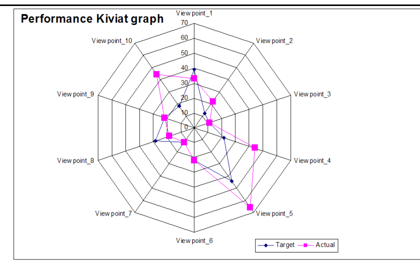
One- and Two–dimensional Graphs	Example
Trend graphs rely primarily on historical time series data to predict the future. Using a trend graph highlights a negative performance trend or makes it possible to capitalize on opportunities based on a positive performance trend.	
Kiviat graphs are circular graphs in which several different performance measures can be plotted along radial lines inside a polygon. Using predetermined scales, the center can be considering the lowest level of performance.	

Table 3.6 Visualizing performance: Multidimensional graphs

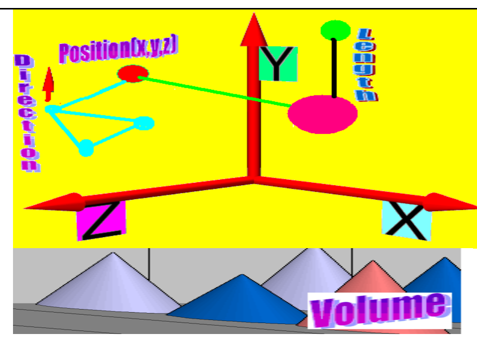
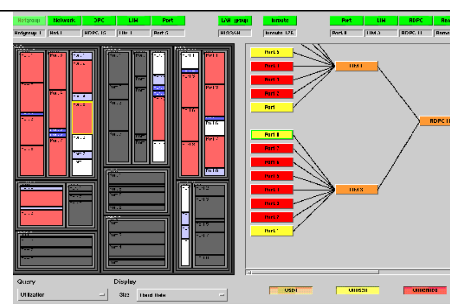
Multidimensional Graphs	Examples
Geometrical modeling visualization creates an image using position, length, direction, area, and volume from different performance variables, and develops a concept around this geometric relationship. “A picture can express ten thousand numbers or ten thousand gigabytes” (Sokolowski and Banks, 2009, p. 104).	
TreeMaps are represented by a series of nested rectangles.	

Table 3.6 Visualizing Performance: Multidimensional graphs

Hierarchical graphs represent a ranking of measures or performances, and each item is subordinate to only one other item. **Tree graphs** are easy to understand and can represent many levels of indicators.	

3.1.4 Performance improvement

It is important that there be a clear link between measures and improvement goals. Performance improvement is achieved by benchmarking, feedback, and planning, as shown in Figure 3.14.

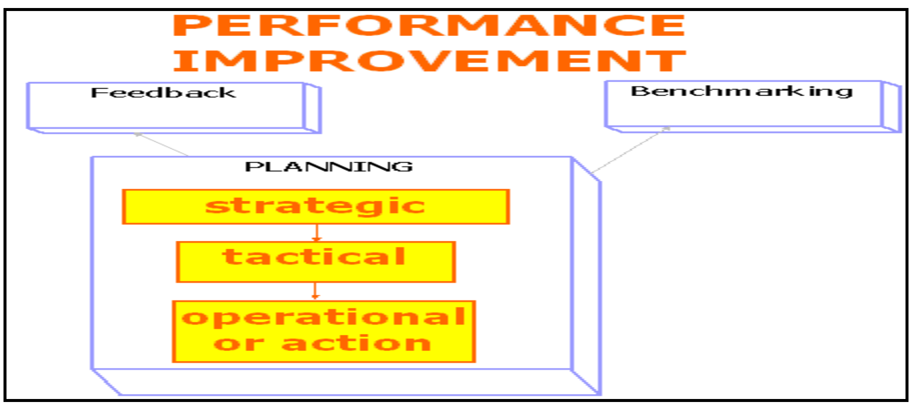


Figure 3.14 Performance improvement.

Benchmarking

Benchmarking is often perceived as a quality initiative, and is an integral part of performance improvement, providing a mechanism for making comparisons in order to better situate organizational performance. The role of the benchmarking must be to direct the actions and organizational changes necessary for the achievement of competitive advantage.

There are four types of benchmarking activities: internal – comparing the performance of various units within the same organization; functional – comparing similar business functions in different organizations; competitive – focusing on direct competitors; and generic – different industries that represent the "best in class" (Holloway *et al.*, 1995).

The International Software Benchmarking Standards Group (ISBSG) database of software project data offers an interesting avenue for benchmarking activities.

Feedback

Feedback must be accepted to be effective in changing behavior. It should be frequent, consistent, and specific, rather than general. For example, day-to-day or continuous feedback is preferable to yearly evaluations. The source of feedback is critical to its acceptance (Mohrman *et al.*, 1989). These authors maintain that it is more likely to be accepted if the source is perceived as being credible, and if it controls important sanctions and rewards (Mohrman *et al.*, 1989).

Planning improvement

It is difficult to make a distinction between conventional strategic planning and performance improvement planning, as shown in Figure 3.14. The definition of strategic, tactical, and operational planning, as described in Table 3.4 and as shown in Figure 3.14, will vary with the organization and also perhaps with the department (Sink and Tuttle, 1989).

According to Holloway and Lewis, there are four fundamental questions to ask when attempting to improve performance (Holloway *et al.*, 1995):

- Are we performing better than before?
- Are we performing better than other business units in the company?
- Are we performing better than our competitors?
- Are there any other industries that are performing well and from whom we can learn?

Important points to take into consideration in performance improvement are:

- The key indicators to be improved must be selected and a well-defined process improvement objective established.
- There are no limitations on how many times the team can attempt to improve performance. If the target has been reached too easily, then new targets must be set up (or existing ones modified) in order to improve performance.
- Any difference between actual performance and target performance is an opportunity for performance improvement.
- If previous objectives are not reached at all, or achieved with great difficulty, it may be necessary to re-adjust one or more targets.

3.1.5 Conclusion

In this chapter, a multidimensional PMF was proposed and described, as shown in the Phase 2 column in Figure 3.1. Four steps were presented, as shown in Figure 3.15:

- Step 1: design of the PMF, in section 3.1.1, an important phase in which the measures are selected and the linkages between them are created;
- Step 2: implementation of measures, in section 3.1.2;
- Step 3: use of the framework to transform data into information and use the appropriate tools to better understand the results in order to facilitate interpretation and decision making, in section 3.1.3;
- Step 4: performance improvement, in section 0, for which proper feedback is important, and of which benchmarking is an integral part.

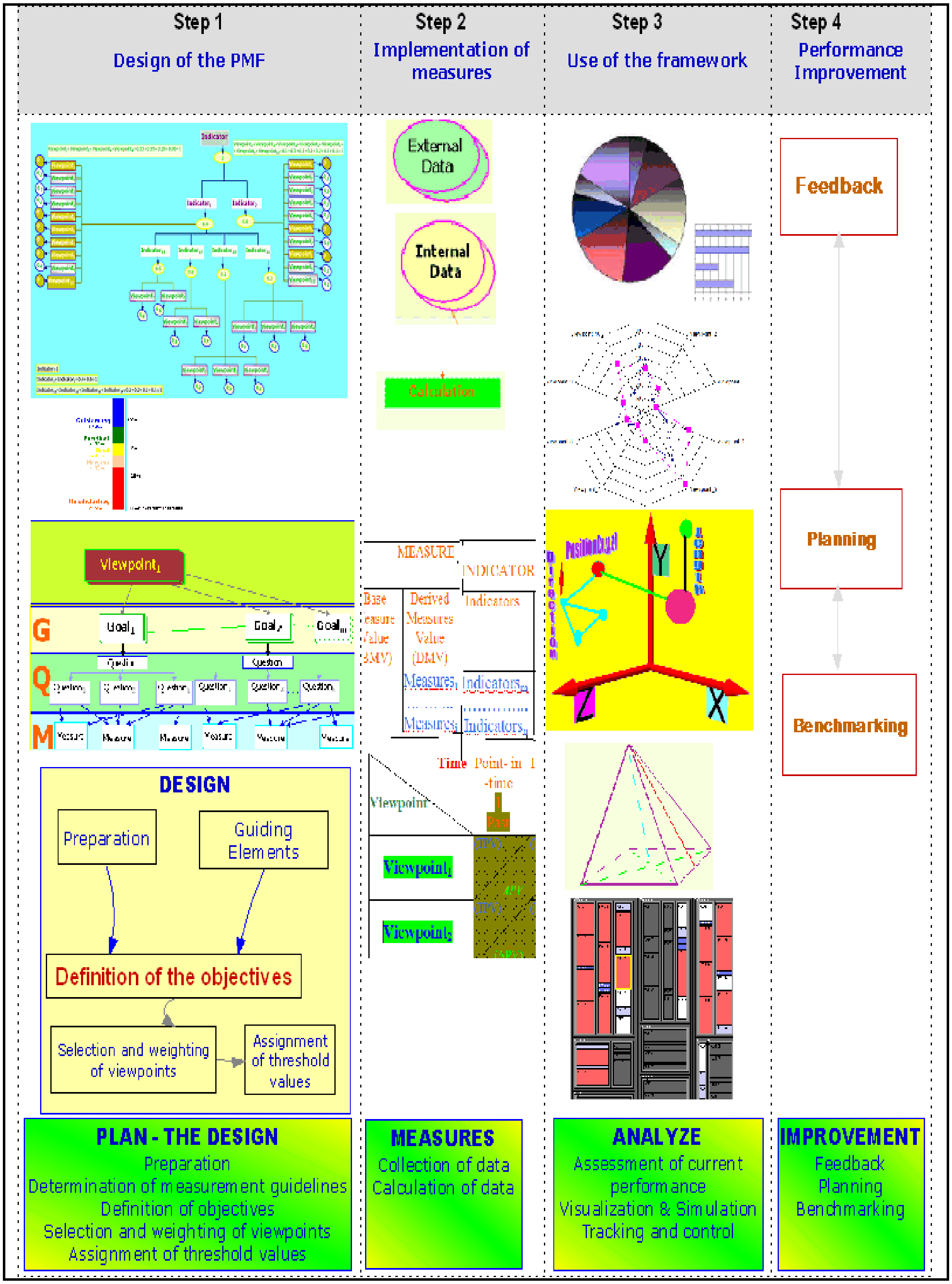


Figure 3.15 Framework summary.

CHAPTER 4

THE MULTIPERF PROTOTYPE

The purpose of this chapter is to present and discuss the software prototype that has been developed, as shown in Figure 4.1, which is based on the conceptual framework discussed in chapter 3.

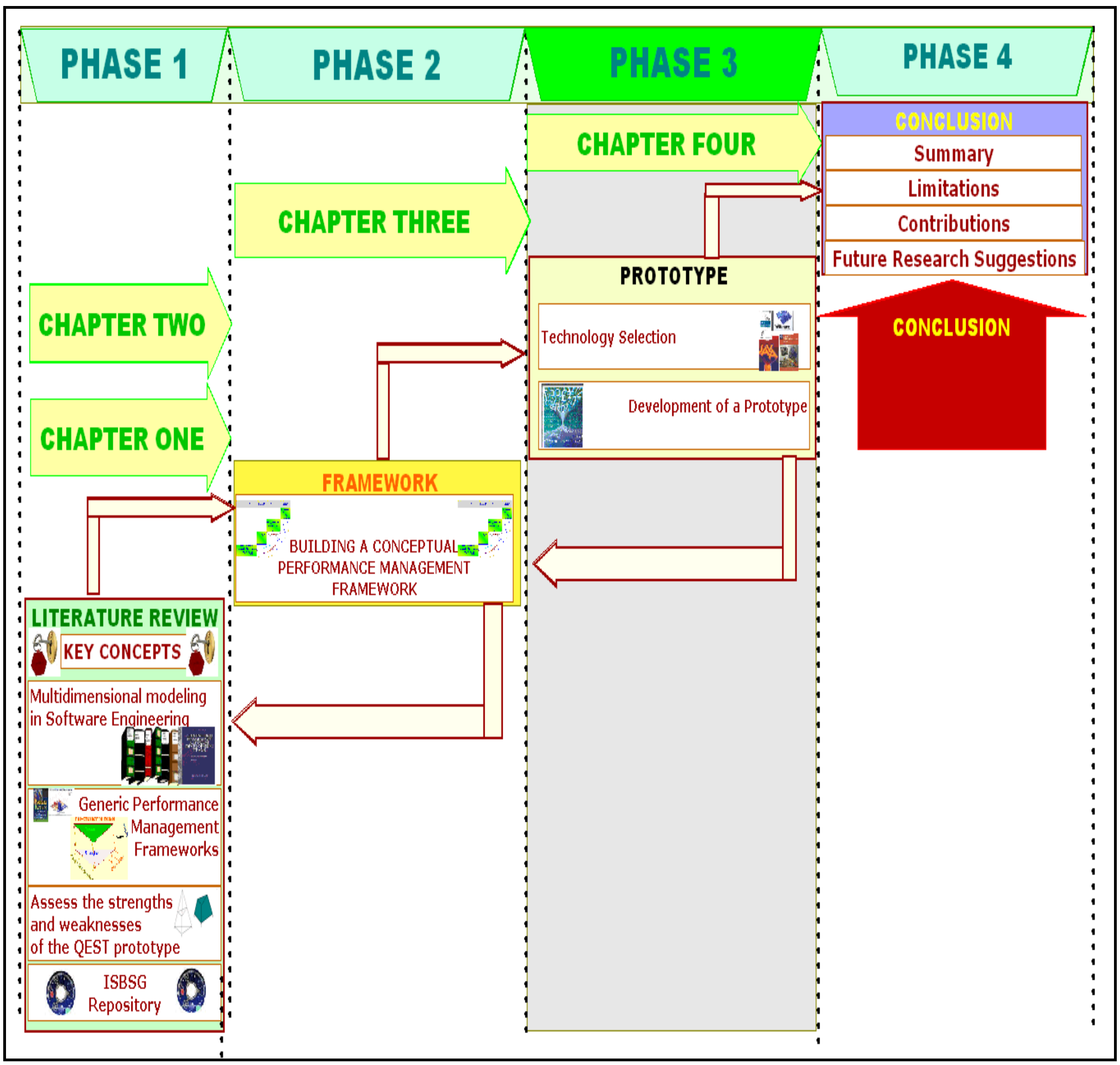


Figure 4.1 Thesis Map – chapter 4 – Development of a prototype.

The chapter is divided into five sections:

- Section 4.1 presents an overview of the technology used to develop the prototype;
- Section 4.2 shows how the Plan-Design step is implemented in the prototype;
- Section 4.3 shows how the Implementation of measures step is incorporated into the prototype;
- Section 4.4 presents the features that are available in the prototype to analyze and model performance data;
- Section 4.5 describes the features that are available in the prototype to analyze the content of the ISBSG Repository, in order to help define performance targets.

The MultiPERF features are presented and discussed as shown in Figure 4.2. On the main screen, the title and the number of the section where the relevant features are found are provided:

- Navigation using a performance tree is explained in section 4.4.1;
- Identification is explained in section 4.2.1;
- Basic reports and graphical representations are described in section 4.4.2;
- Visual analytics and statistical analysis are described in section 4.4.3 and in section 4.5;
- Tracking and selecting indicators and measures are presented in section 4.4.1 and in section 4.4.2;
- Functionalities relating to the ISBSG database are analyzed in section 4.5.

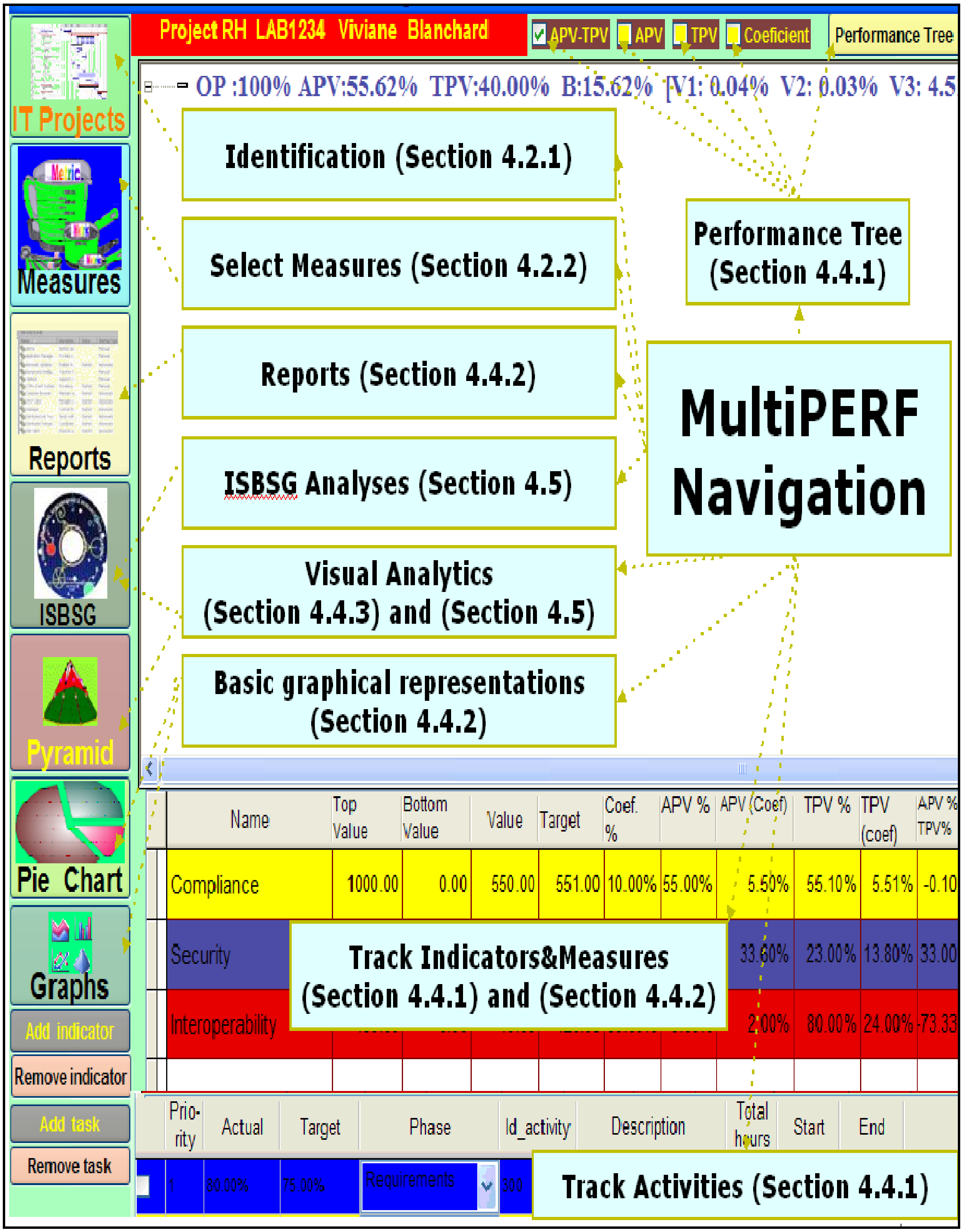


Figure 4.2 MultiPERF – main figure.

4.1 Technology selection

How was the adopted technology chosen? First of all, the ISBSG Repository is itself an Excel spreadsheet. The use of Visual Basic Application[1] is therefore a logical choice for representing the various models graphically. Then, there is the need for a database management system to store the performance data.

The Visual FoxPro (VFP)[2] programming environment, including the human-machine interface, was adopted, mainly because:

- it includes its own relational database engine;
- it supports SQL queries and data manipulations;
- it includes a full-featured programming language; and
- the author has considerable experience with this environment.

Finally, OpenGL was used for the more advanced graphical presentations. According to www.opengl.org, OpenGL is the 'most widely used and supported 2D and 3D graphics application programming interface (API)'[3]. It was selected because there is a need to specify graphical objects and their attributes, as well as to manipulate images.

4.2 The design of MultiPERF

This section describes the way in which the MultiPERF prototype implements the Plan-Design step, as shown in Figure 4.3. This section is divided into four subsections, which are very clearly linked to the conceptual framework presented in chapter 3 as follows:

---

[1] http://msdn.microsoft.com/en-us/library/bb190882%28v=office.11%29.aspx

[2] http://msdn.microsoft.com/en-us/vfoxpro/default.aspx

[3] http://www.opengl.org/about/overview/

- Section 4.2.1: Setting up a Performance Management Team which implements the concepts presented in section 3.1.1.1 and Determination of Guiding Elements which implements the concepts presented in section 3.1.1.2;
- Section 4.2.2: Definition of objectives which implements a first subset of concepts presented in section 3.1.1.3;
- Section 4.2.3: Section Selection and weighting of measures and viewpoints which implements the concepts presented in section 3.1.1.4;
- Section 4.2.4: Assignment of threshold values which implements the concepts presented in section 3.1.1.5.

In the next chapter, as well, section 4.5.1: Setting performance targets implements a second subset of concepts presented in section 3.1.1.3.

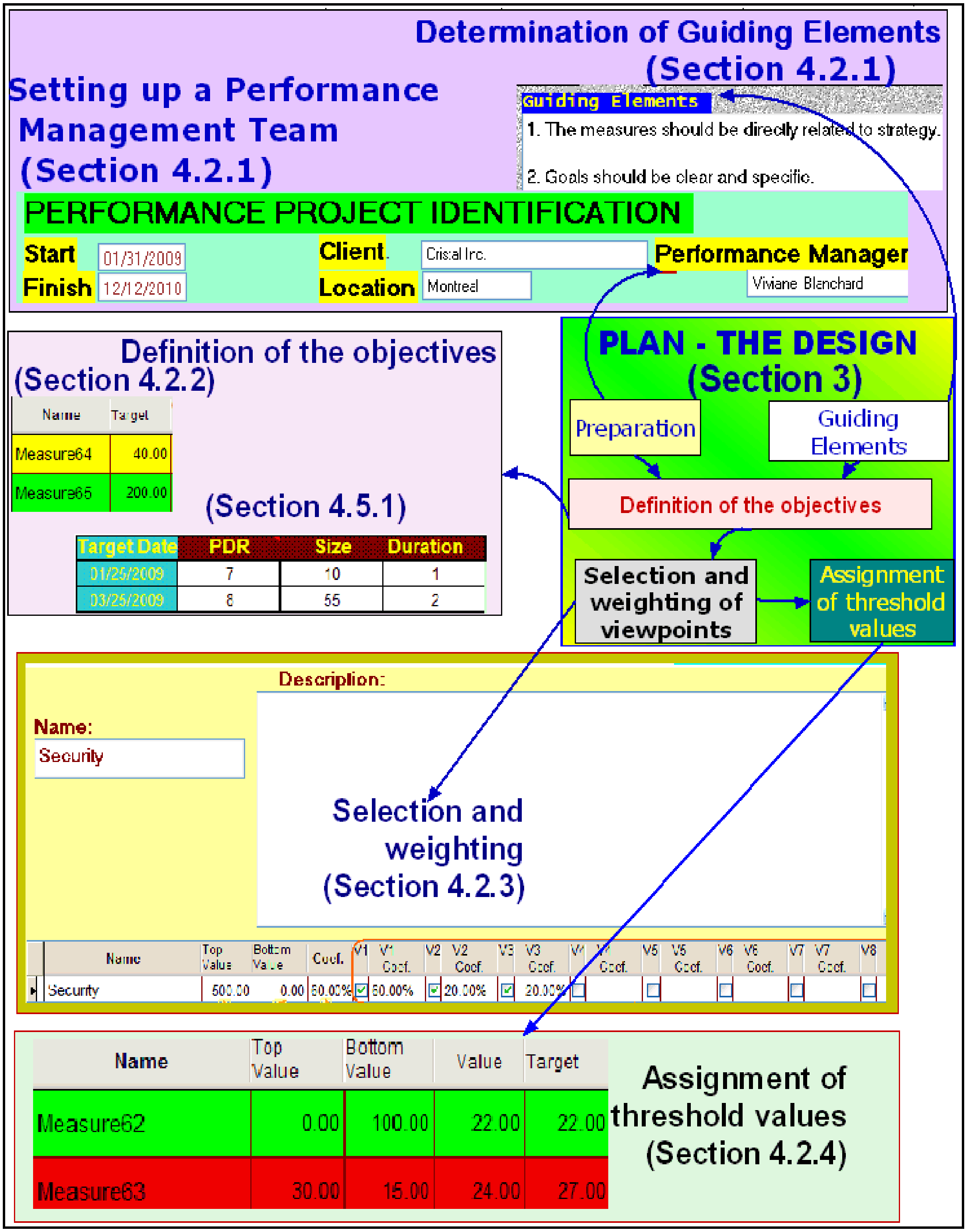


Figure 4.3 Overview of the step: Plan - The design in MultiPERF.

4.2.1 Setting up a performance management team and determination of guiding elements

An organization should set up a performance team to manage organizational performance. The team will define and select the measures and viewpoints necessary to accurately represent performance and to enable its continuous improvement.

Figure 4.4 shows a screen snapshot shows of how information is entered into the MultiPERF prototype with respect to guiding elements. The person responsible for performance management (Performance Manager), can enter on this screen some basic information on the performance improvement project.

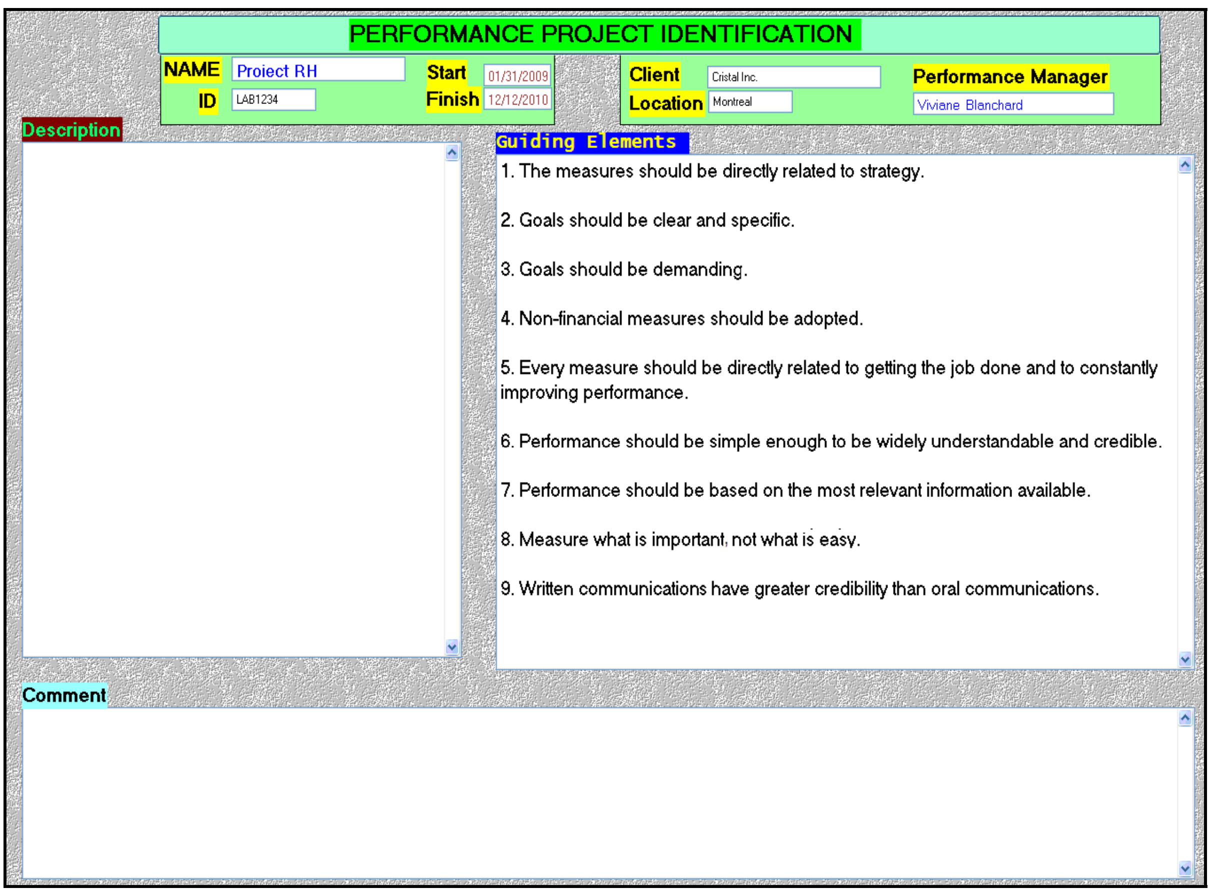


Figure 4.4 Performance project identification and guiding elements.

4.2.2 Definition of the objectives

Performance is a multidimensional concept – distinct, but related, viewpoints (each viewpoint representing a distinct dimension of performance) must be taken into account simultaneously.

Organizational goals define the desired level of performance and represent the objectives of the organization as it was described in the section 3.1.1.3. The TPV for each indicator and measure is defined in the prototype, and each viewpoint (see 4 in Figure 4.5) is decomposed into a set of goals (see 1 in Figure 4.5) and questions (see 2 in Figure 4.5) are associated with each measure (see 3 in Figure 4.5), as shown in an example in Figure 4.5. In the next chapter, the process of setting targets is described in the section 4.5.1 using an example.

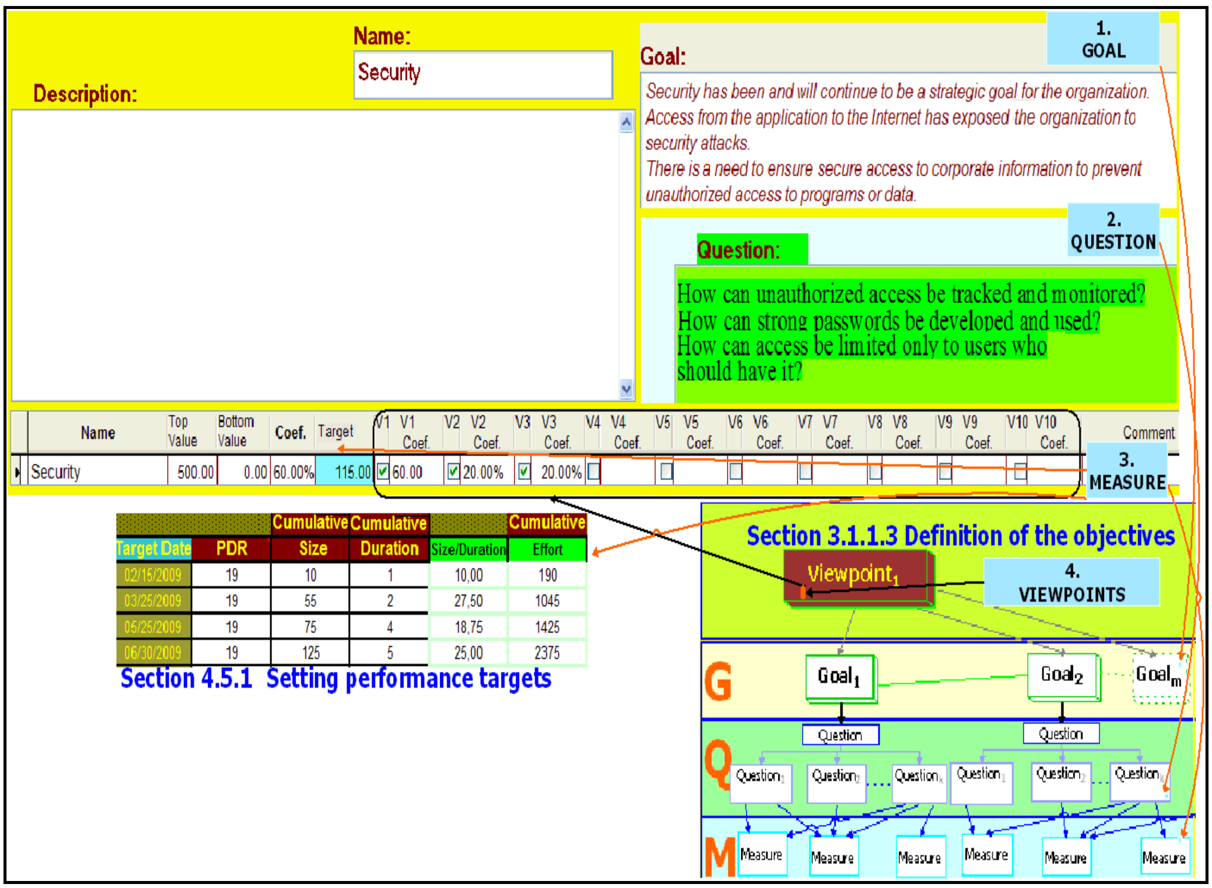


Figure 4.5 Definition of objectives.

4.2.3 Selection and weighting of measures and viewpoints

Viewpoints are selected from a list of viewpoints, and this is achieved using check boxes. As shown in Figure 4.6 and Figure 4.16, a number is associated with each viewpoint to facilitate visual display (for example, in Figure 4.6, V_1=Economic, V_3=Technical, V_4=Profitability, and V_{10}=Quality of Work Life). It is also possible to select the entire set or only a subset of the viewpoints, as shown in Figure 4.6 and Figure 4.16.

VIEWPOINTS

Viewpoint	Name	Selected
1	Economic	☑
2	Social	☐
3	Technical	☑
4	Profitability	☑
5	Effectiveness	☐
6	Efficiency	☐
7	Innovation	☐
8	Quality	☐
9	Productivity	☐
10	Quality of Work Life	☑
11	Other1	☐
12	Other2	☐
13	Other3	☐
14	Other4	☐
15	Other5	☐
16	Other6	☐
17	Other7	☐
18	Other8	☐
19	Other9	☐
20	Other10	☐
21	Other11	☐
22	Other12	☐
23	Other13	☐
24	Other14	☐
25	Other15	☐

Figure 4.6 Select viewpoints.

Measures can be selected in MultiPERF from a predefined list of measures, and can be combined in different ways based on indicators, as shown in Figure 4.7, where:

- The initial predefined list of measures proposed by the prototype comes from the ISO-9126 (ISO/IEC, 2001) standard. This list can be modified by the user by the addition or deletion of measures.
- The indicators and measures are then selected. For example, the Internal Metrics and External Metrics indicators were selected in scenario A, but not Quality in Use. In scenario B, Internal Metrics, External Metrics, and Quality in Use were all selected.
- A verification of the sum of the weights of the children of any node, which must be exactly 100%, is completed to validate the selection. For example, in Figure 4.7:
  - Scenario A: Internal Metrics (70%) + External Metrics (30%) =100%=1
  - Scenario B: Internal Metrics (70%) + External Metrics (10%) + Quality in Use (20%) =100%=1.

At a lower level, the weight of the children of Internal Metrics: Functionality (5%) + Usability (5%) + Efficiency (5%) + Maintainability (5%) + Reliability (10%) + Portability (70%) =1. The weight of the children of Functionality is: Compliance (10%) + Security (60%) + Interoperability (30%) =100%=1.

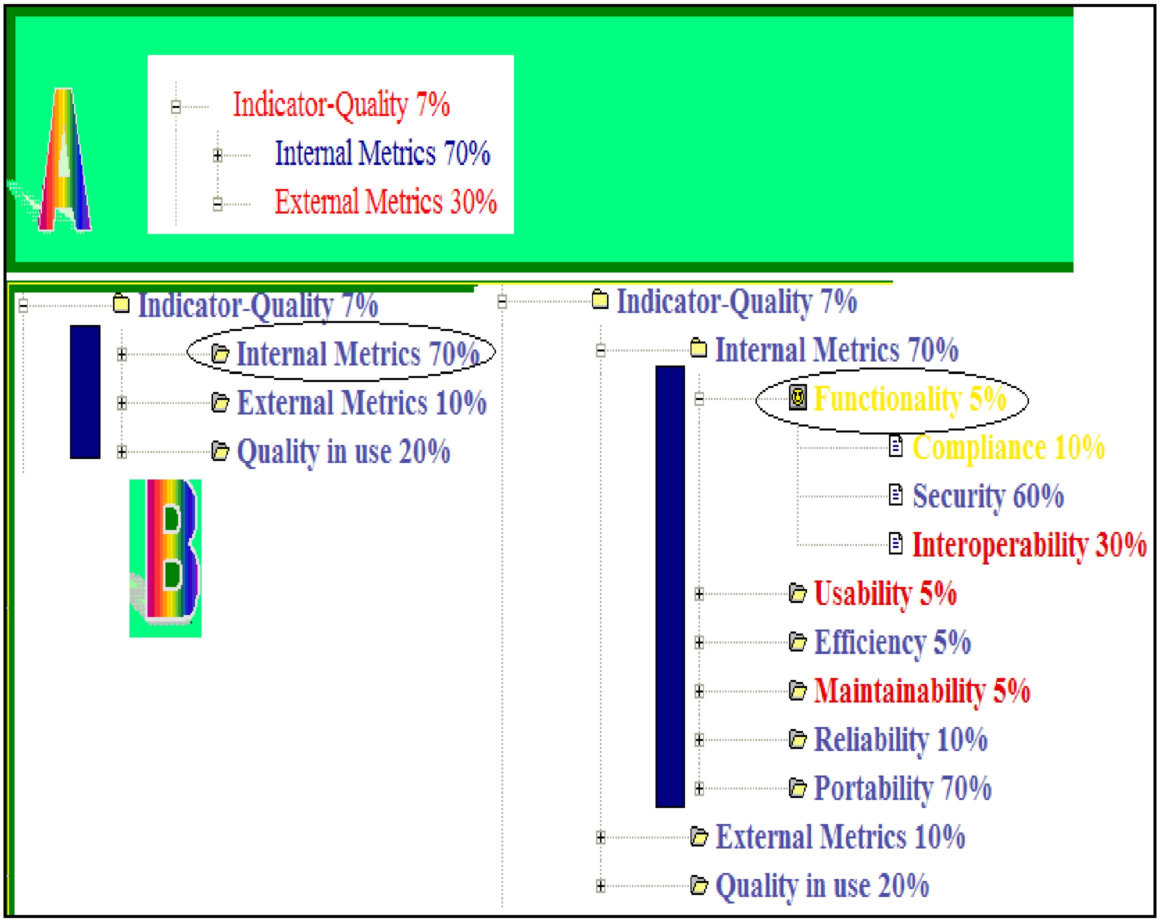


Figure 4.7 Selection and weighting of indicators and measures.

Every measure has one or more viewpoints associated with it, as shown in Figure 4.8, where the Security measure is associated with: V_1(Economical) 60.00% + V_2(Social) 20.00% + V_3(Technical) 20.00% = Total Value (100%). The difference between one and the sum of the weights of the selected viewpoints must be always zero.

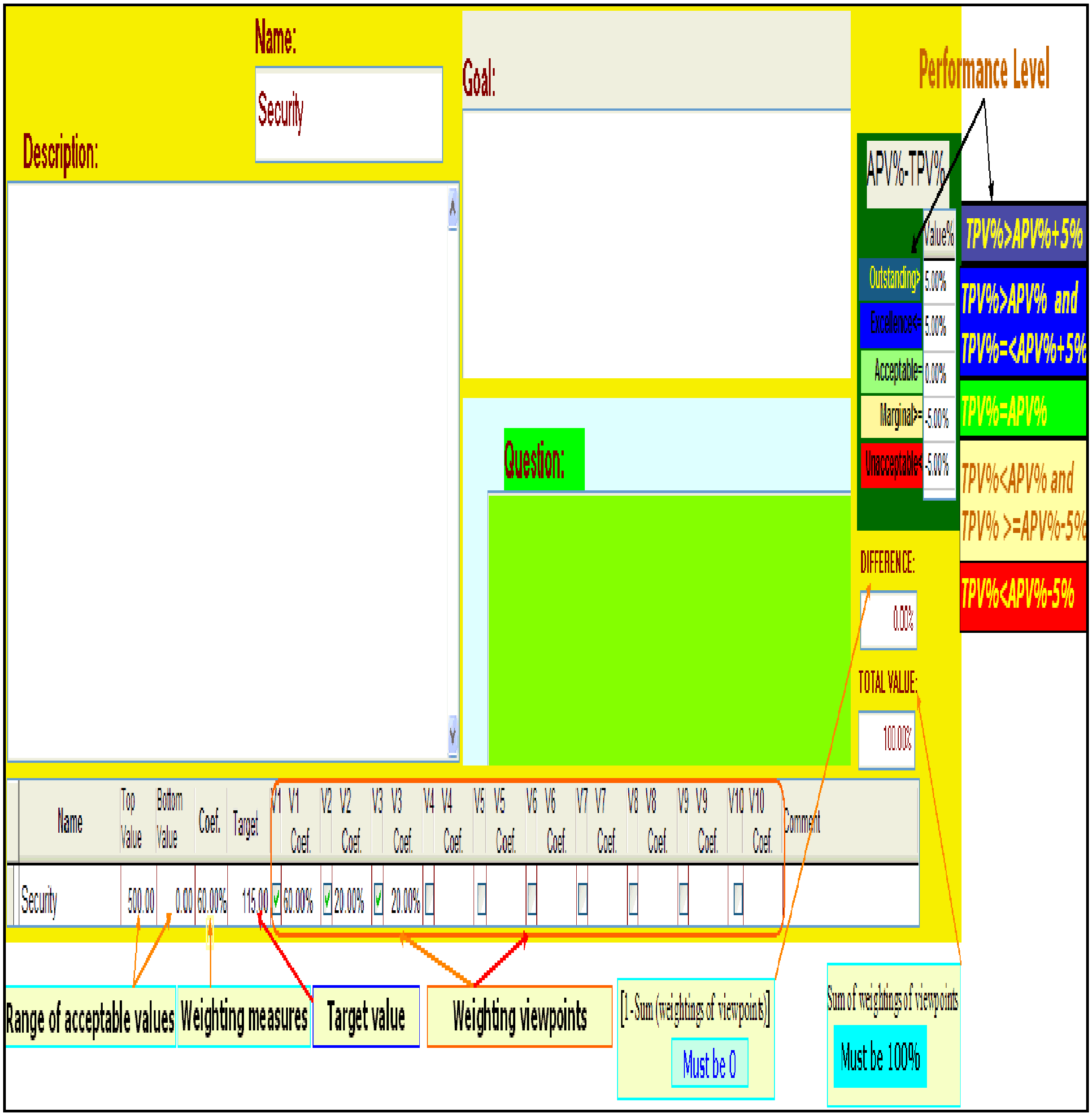


Figure 4.8 Weighting of measures and assignment of threshold values.

4.2.4 Assignment of threshold values

The manager must assign threshold values by defining a range of acceptable values for each measure, as shown in Figure 4.8:

- The lower and upper threshold values, represented by the top value and the bottom value, can be different from one measure to another. For example, acceptable values for Security must be between 0 (bottom value) and 500 (maximum value), as shown in Figure 4.8.
- All the values that are within the acceptable range will automatically be normalized on a scale from zero to one.
- Performance level: comparing target performance values (TPV) with actual performance values (APV) is explained in section 4.3.3.

Table 4.1 shows an example of how a performance gap will be calculated. The columns in Table 4.1 are:

- Top Value - the top performance value for the measure;
- Bottom Value – the lowest performance value for the measure. Note that the bottom value can be greater than the top value if the scale is inverted; for example, on a scale of one to five, the top performance value could be one or it could be five;
- Value – the value obtained for the measure before normalization;
- Coef - weight assigned to the measure. This indicates the importance of each measure using a scale from zero (0% = 0 points = minimum) to one (100% = 100 points = maximum);
- Performance :
  - Calculate APV% and APV(coef) for Measure64, Measure63, and Measure62, as shown in Figure 4.9
  - APV% = Actual Performance Value = actual performance value for the measure after normalization;
  - APV(coef) = APV% x Coef.

Table 4.1 Example of how the APV% and APV(Coef) will be calculated

Top Value	Bottom Value	Value	Coef	Performance
50	0	20	5%	$APV\% = \frac{(\text{Value}) - (\text{Bottom\_Value})}{(\text{Top\_Value}) - (\text{Bottom\_Value})} * 100\% =$ $= \frac{20 - 0}{50 - 0} * 100\% = 40\%$ $APV(Coef) = Coef * 40\% = 5\% * 40\% = 2\%$
30	15	20	90%	$APV\% = \frac{(\text{Value}) - (\text{Bottom\_Value})}{(\text{Top\_Value}) - (\text{Bottom\_Value})} * 100\% =$ $= \frac{20 - 15}{30 - 15} * 100\% = 33.33\%$ $APV(Coef) = Coef * 33.33\% = 90\% * 33.33\% = 30\%$
0	10000	20	5%	$APV\% = \frac{-(\text{Value}) + (\text{Bottom\_Value})}{(\text{Bottom\_Value}) - (\text{Top\_Value})} * 100\% =$ $= \frac{-20 + 10000}{10000 - 0} * 100\% = 99.8\%$ $APV(Coef) = Coef * 99.80\% = 5\% * 99.80\% = 4.99\%$

Name	Top Value	Bottom Value	Value	Coef. %	APV %	APV (Coef)
Measure62	0.00	10000.00	20.00	5.00%	99.80%	4.99%
Measure63	30.00	15.00	20.00	90.00%	33.33%	30.00%
Measure64	50.00	0.00	20.00	5.00%	40.00%	2.00%

Figure 4.9 MultiPERF performance grid for Table 4.1.

4.3 Implementation of measures

The implementation of measures starts with the collection of internal data, proceeds to the use of the external data available, and finishes with the step called Calculate Performance Data, as shown in Figure 4.10. This section explains how the MultiPERF prototype enables the implementation of measures step.

4.3.1 Internal data

In the prototype, no internal data are available for any organization, and so the internal data shown in this thesis are for demonstration purposes only. For all the internal data contained in Figure 4.4, Figure 4.6, and Figure 4.8, as well as for the target value in Figure 4.11, the prototype contains the usual data entry, edit, and storage functions.

4.3.2 External data

The only external data currently available in the prototype are those in the ISBSG Repository, which was presented in section 2.4. These data are stored in an Excel spreadsheet file when initially received from the ISBSG organization. In section 4.5, the MultiPERF prototype features that are based on the ISBSG Repository are presented, and how to use this repository to set targets and desired performance levels is explained.

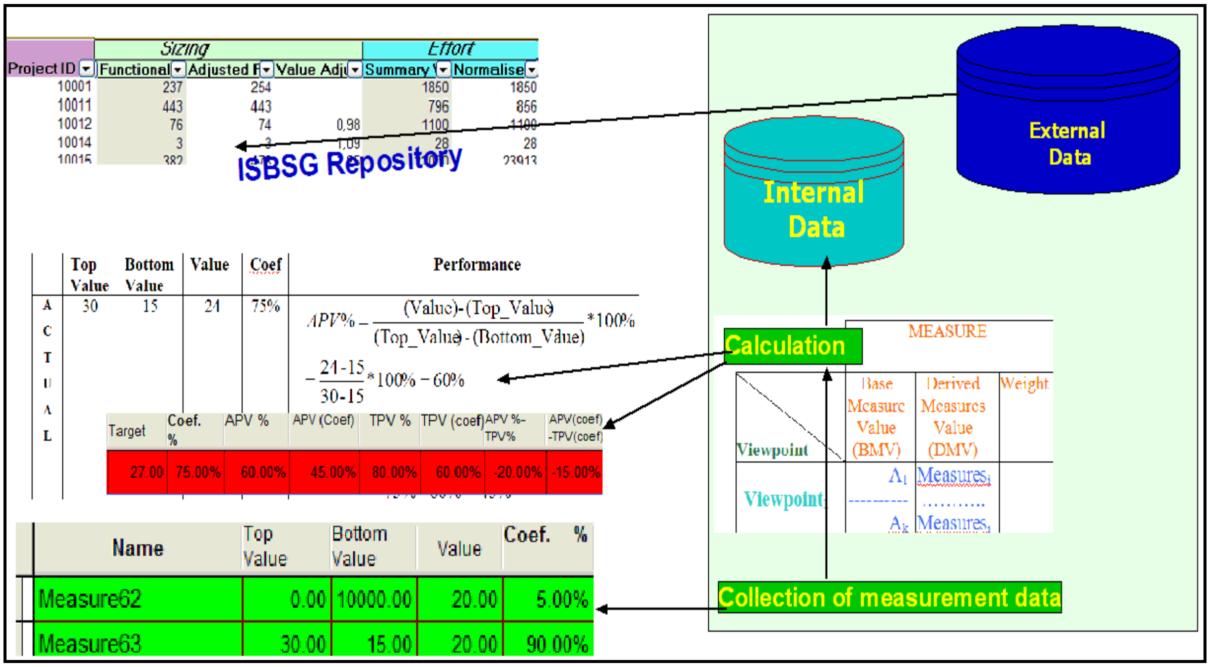


Figure 4.10 Implementation of measures.

4.3.3 Calculate performance data

Comparing target performance values (TPV) with actual performance values (APV) is essential to managing organizational performance. The MultiPERF prototype considers differences between APV and TPV in the following manner:

- When the APV% is greater than the TPV% by 5%, performance is considered outstanding. If it is less than 5%, but still more than 0%, performance is excellent.
- When the APV% and the TPV% are equal, performance conforms to the plan, and is therefore considered to be good (acceptable);
- When the APV% is less than 5% below the TPV%, performance, performance is considered marginal (but still acceptable);
- When the APV% is more than 5% below the TPV%, performance is unacceptable.

The minimal or lowest performance value is always zero, and the maximum or highest performance value that can be obtained will always be one.

Indicators are not necessarily equally important – the role of the weighting indicators is to assign an importance to each indicator as viewed by the manager. For example, in Scenario B of Figure 4.7, security is assigned a weight 6 times higher in calculating performance than compliance, and twice as high as interoperability.

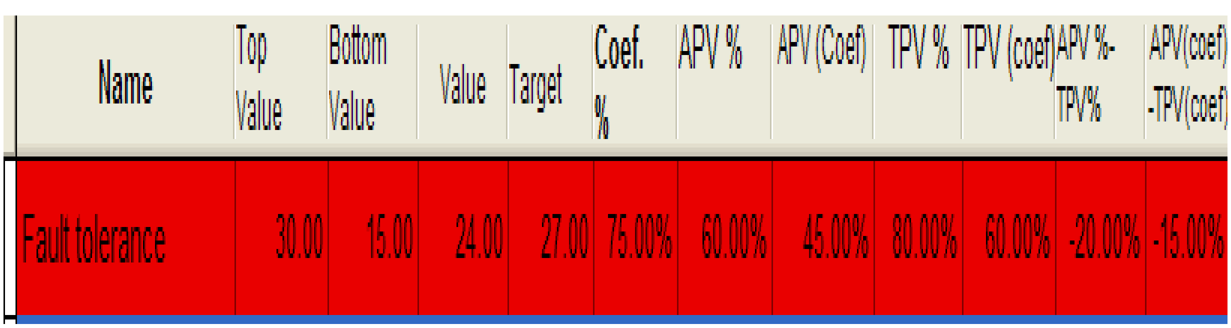
An example of how a performance gap will be calculated for Fault Tolerance is shown in Table 4.2 and Figure 4.11. Columns in Table 4.2 are the following:

- Top Value – the top performance value for the measure;
- Bottom Value – the lowest performance value for the measure;
- Value (and Target as is shown in Figure 4.11) – the value of the measure before normalization;
- Coef – the weight assigned to the measure;
- Performance:
  - APV% = Actual Performance Value = actual performance value for the measure after normalization;
  - TPV% = Target Performance Value = target performance value for the measure after normalization;
  - APV(coef) = actual performance value for the measure after taking into account the coefficient assigned to the measure;
  - TPV(coef) = target performance value for the measure after taking into account the coefficient assigned to the measure.

The grid, as shown in Figure 4.11, is colored red, because the performance gap for the measure presented in Table 4.2 between the APV% and the TPV% is unacceptable: APV%-TPV%=-20%. The meaning of the colors is explained in section 4.4.1

Table 4.2 Example of how the performance gap will be calculated

	Top Value	Bottom Value	Value	Coef	Performance
ACTUAL	30	15	24	75%	$APV\% = \frac{(\text{Value}) - (\text{Top\_Value})}{(\text{Top\_Value}) - (\text{Bottom\_Value})} * 100\% = \; = \frac{24 - 15}{30 - 15} * 100\% = 60\%$ $APV(coef) = Coef * APV\% = \; = 75\% * 60\% = 45\%$
TARGET	30	15	27	75%	$TPV\% = \frac{(\text{Value}) - (\text{Top\_Value})}{(\text{Top\_Value}) - (\text{Bottom\_Value})} * 100\% = \; = \frac{27 - 15}{30 - 15} * 100\% = 80\%$ $TPV(coef) = Coef * TPV\% = \; = 75\% * 80\% = 60\%$
				Difference **APV-TPV**	The performance gap for this measure is: APV % - TPV% = 60% - 80% = -20% APV(coef) - TPV(coef) = 45% - 60% = -15%



Name	Top Value	Bottom Value	Value	Target	Coef. %	APV %	APV (Coef)	TPV %	TPV (coef)	APV %-TPV%	APV(coef) -TPV(coef)
Fault tolerance	30.00	15.00	24.00	27.00	75.00%	60.00%	45.00%	80.00%	60.00%	-20.00%	-15.00%

Figure 4.11 MultiPERF performance grid for Table 4.2.

The sum of the APVs for the *m* measures linked to Viewpoint_1, as shown in Figure 4.12, is the APV for Viewpoint_1:

$$\text{APV(Viewpoint\_}_1) = \sum_{k=1}^{m} APV(Measure\_{}_k) * \text{Weight}(Measure\_{}_k(Viewpoint\_{}_1))$$

The OP_APV or Organizational Actual Performance (as seen in the left-top corner of Figure 4.12) is the sum of the APVs for the *n* viewpoints, as shown in Figure 4.12:

$$\text{OP\_APV} = \sum_{i=1}^{n} APV(Viewpo\text{int\_}_i)$$

The sum of the TPVs for the *m* measures linked to Viewpoint_1 is the TPV for Viewpoint_1:

$$\text{TPV(Viewpoint\_}_1) = \sum_{k=1}^{m} TPV(Measure\_{}_k) * \text{Weight}(Measure\_{}_k(Viewpoint\_{}_1))$$

The sum of the TPVs for the *n* viewpoints is the OP_TPV, or Organizational Target Performance TPV for the organization:

$$\text{OP\_TPV} = \sum_{i=1}^{n} TPV\,(Viewpoint\ \_{}_i)$$

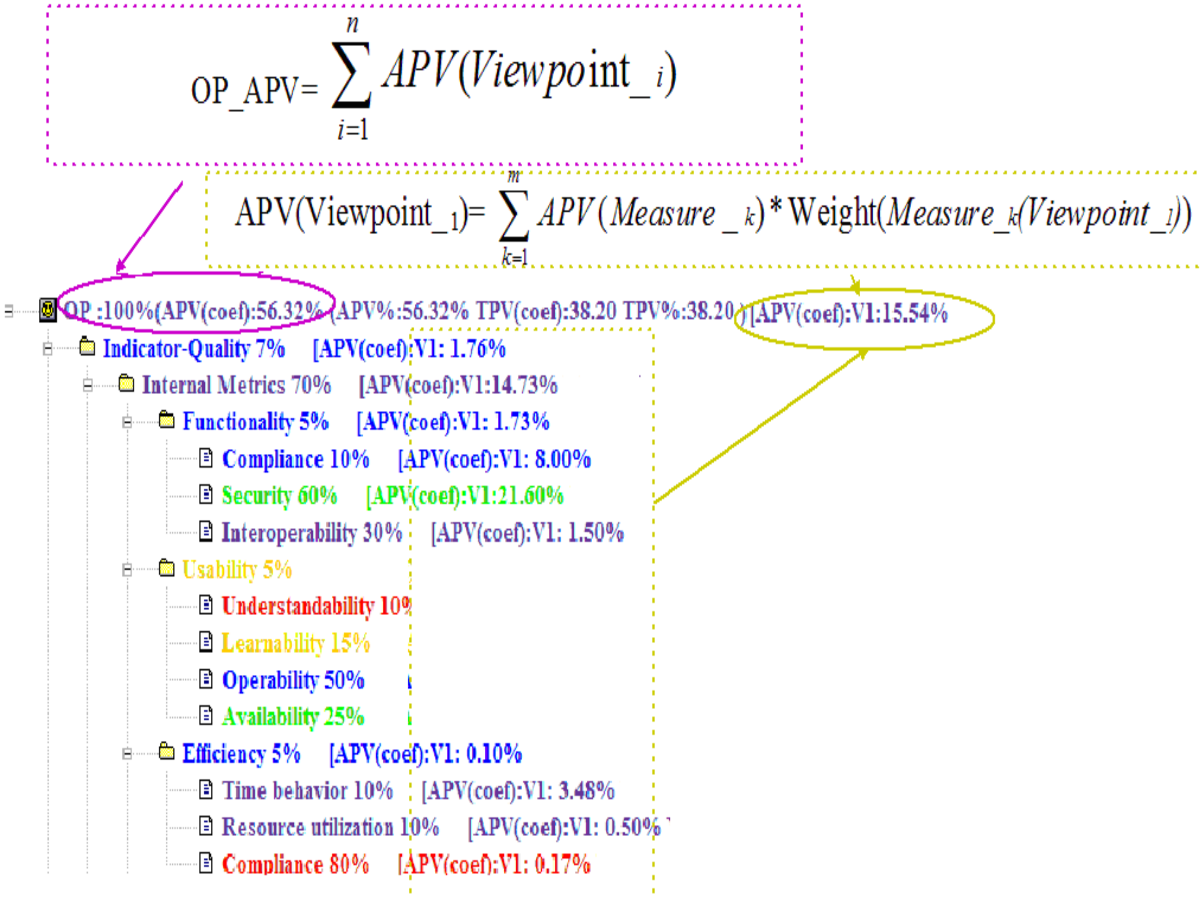


Figure 4.12 Calculating APV.

4.4 Analysis and use of performance data

Analyzing, understanding, and explaining what has occurred in an organization in a particular context using performance data is important for quality decision making.

The actual performance values (APVs) are the values as measured in the organization. The target performance values (TPVs) are the values assigned by the performance manager. Performance is analyzed essentially by comparing APV to TPV.

Measures have now been selected, weightings and threshold values have been assigned, as described in section 4.2.2, section 4.2.3, and section 4.2.4, the measures have been collected and implemented, as shown in section 4.3, and the APVs and TPVs have been automatically calculated.

In this section, all this information is modeled and represented graphically, in order to be better understood and analyzed by managers. The basic features are described briefly, and the more advanced analytical tools are presented in more detail. This section is divided into three subsections:

- Building a performance tree and a performance grid, and highlighting and using color in tracking performance (section 4.4.1);
- Basic reports and the use of basic graphical representations to analyze measures, indicators, and viewpoints (section 4.4.2);
- Building the pyramid (section 4.4.3).

4.4.1 Performance tree and use of color

The performance tree, as shown in Figure 4.13, Figure 4.14, and Figure 4.15, is linked to a performance grid, as shown in Figure 4.16, and each node in the tree is linked to a measure or an indicator. Drill paths are a straightforward way to retrieve information in the context of performance management. The MultiPERF prototype contains a feature for drilling up and drilling down indicators and measures by clicking on the performance tree, and also choosing different scenarios to be displayed, as shown in Figure 4.13, where some sample scenarios are presented:

- Scenario A shows the weighting of measures, indicators, and viewpoints;
- Scenario B shows the APV%, APV(coef), TPV%, and TPV(coef), B%=APV%-TPV%, B(coef) =APV(coef)–TPV(coef). In the same manner, these figures can be generated for APV(coef), TPV(coef) and for all measures and indicators and the gap between APV, TPV, B%=APV%-TPV%, and B(coef) =APV(coef)–TPV(coef) for each viewpoint and for each measure and indicator.

In addition, an automatically summarized drill-up (collapse all measures and indicators) representation and drill-down (expand all measures and indicators) representation is also available for a performance tree.

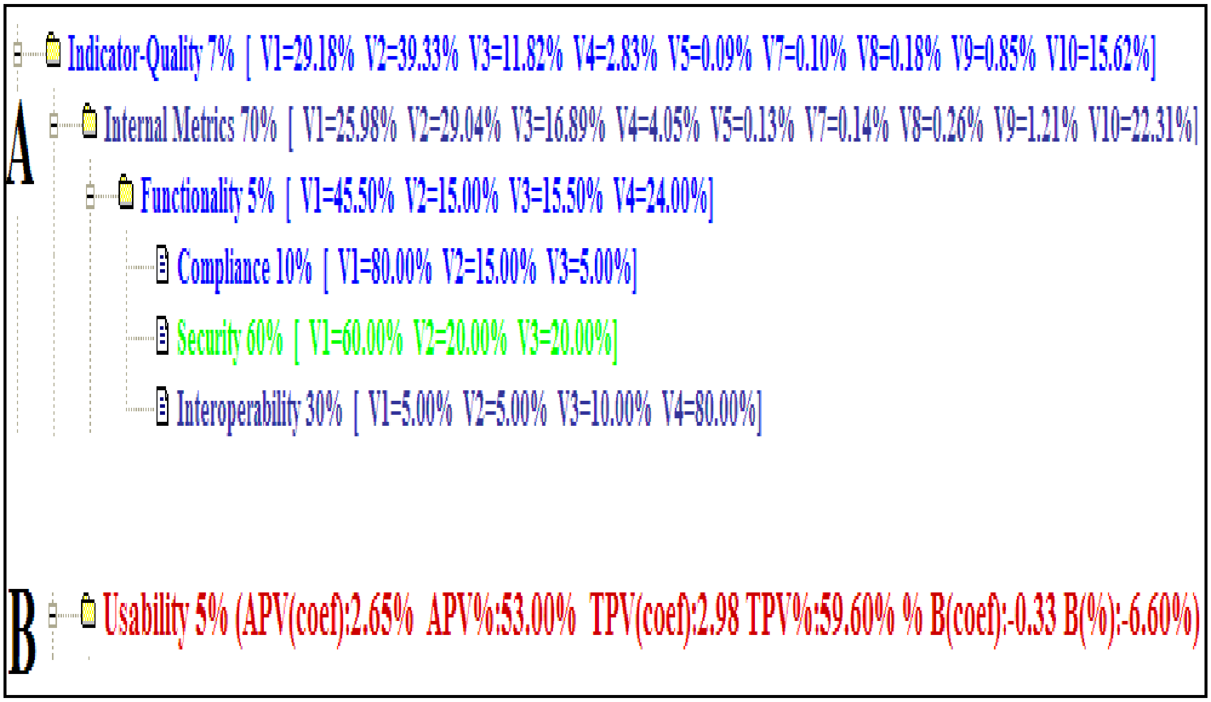


Figure 4.13 MultiPERF performance tree - Snapshot of a few scenarios.

As shown in Figure 4.14, a mapping is created between the conceptual framework described in section 3.1.1.4, the selection and weighting of viewpoints and indicators, and the prototype.

In this figure, selecting or not selecting a performance viewpoint, a measure, or an indicator is illustrated using arrows and letters:

- Legend A identifies the non selected viewpoints in the tree, and results in the absence of those viewpoints for that measure or indicator, as shown using a rectangle in the lower part of the figure, with V_5,V_6,V_7,V_8,V_9 and V_{10} not being selected for Functionality.

- Legend B identifies the selected viewpoints in the tree and results in the presence of those viewpoints for that measure or indicator, as is shown using a rectangle in the lower part of the figure, with V_1,V_2,V_3,V_4 for Interoperability.
- Legend C identifies the measures and indicators selected for $Indicator_1$. This indicator was divided into 3 sub-indicators, and then into measures for each indicator.

Display of a viewpoint means that it was selected for that indicator or measure. The viewpoints are selected only for the lowest level – the upper level will be automatically updated. High and lower levels for indicators are also highlighted. The levels of performance are easy to understand, and are explained in Figure 4.15.

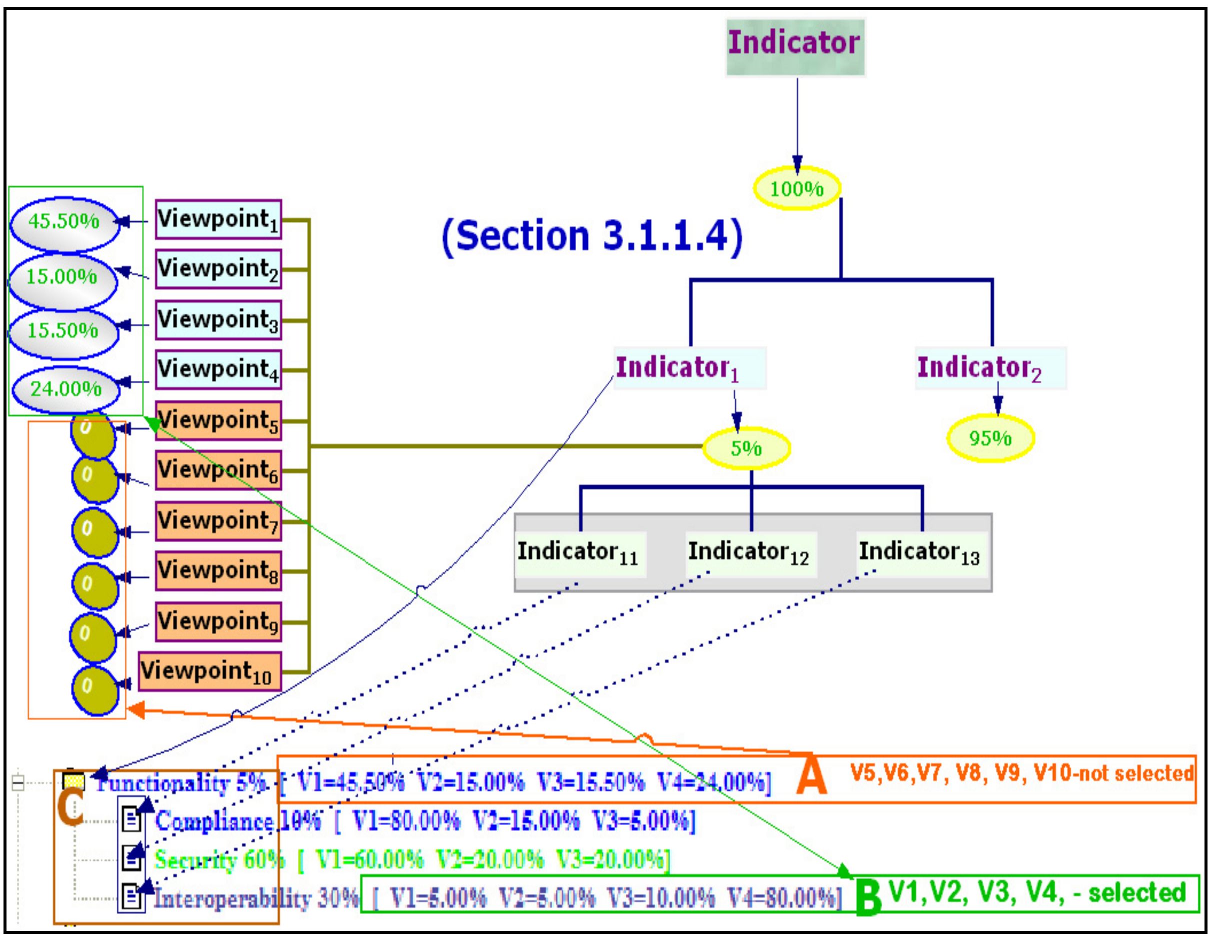


Figure 4.14 MultiPERF performance tree.

A grid is a practical way to enable managers to compare the APV and TPV of measures and indicators, as shown in Figure 4.16, where the color of rows is automatically changed according to the following schema:

- Blue means that performance exceeds expectations. The APV% and TPV% for Security is positive by 33%, and therefore its performance is outstanding, as shown by the navy blue color. Two shades of blue are used, as a function of the percentage: light blue for less than 5%, and navy blue for more than 5%.
- Green means that performance is acceptable;
- Yellow signals a warning, indicating that this measure should be watched. The APV is therefore marginally lower than expected by a non significant amount (by default it is 5%, but this can be changed by the user). For example, the difference between the APV% and the TPV% for Compliance is -0.10%;
- Red means that performance is unacceptable, or that action is needed. For example, the difference between the APV% and the TPV% for Interoperability is -73.33%. The APV is therefore unacceptable for this measure.

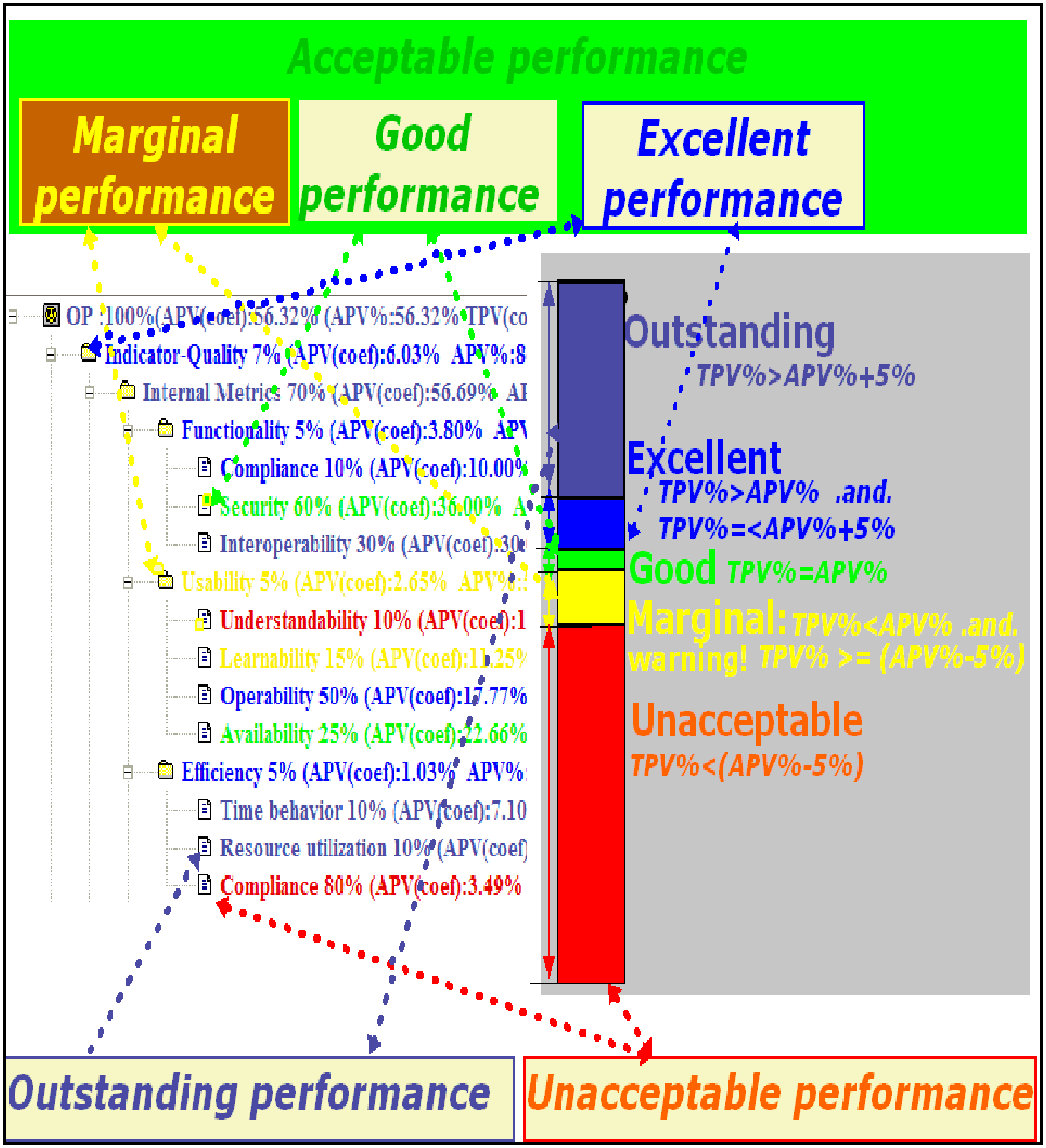


Figure 4.15 Use of colors in a MultiPERF performance tree.

Name	Top Value	Bottom Value	Value	Target	Coef. %	APV %	APV (Coef)	TPV %	TPV (coef)	APV %-TPV%	APV(coef) -TPV(coef)	V1%	V2%	V3%	V4%
Compliance	1000.00	0.00	550.00	551.00	10.00%	55.00%	5.50%	55.10%	5.51%	-0.10%	-0.01%	☑ 80.00%	☑ 15.00%	☑ 5.00%	☐ 0.00%
Security	500.00	0.00	280.00	115.00	60.00%	56.00%	33.60%	23.00%	13.80%	33.00%	19.80%	☑ 60.00%	☑ 20.00%	☑ 20.00%	☐ 0.00%
Interoperability	150.00	0.00	10.00	120.00	30.00%	6.66%	2.00%	80.00%	24.00%	-73.33%	-22.00%	☑ 5.00%	☑ 5.00%	☑ 10.00%	☑ 80.00%

Figure 4.16 MultiPERF performance grid.

It is also possible to analyze each activity separately within a phase, as shown in Figure 4.17:

- analysis can be performed at the activity level of the software life cycle;
- the color of the rows is automatically changed, according to the method explained below, in section 4.4.1, for the MultiPERF Performance Grid and the MultiPERF Performance Tree.
- Priority: a number related to priority is assigned to determine: high, low, or moderate priority;
- Actual: actual performance value for the activity;
- Target: target performance value for the activity;
- Phase: the classical 6-phase waterfall SLC structure are added: requests, specification, design, coding, testing, and maintenance. Other phase (tasks/activities) can be added as well;
- Id_activity and Description: a description and an ID are attributed to each task/activity;
- Planning: a schedule for conducting the activity, including a start date and an end date;
- Individual in charge (responsible): determines the resources needed to accomplish the activity;

	Prio-rity	Actual	Target	Phase	Id_activity	Description	Total hours	Start	End	Responsible
☐	1	80.00%	75.00%	Requirements	300		15	02/05/10	02/05/10	Eve Vazi
☐	2	85.00%	86.00%	Specification	400		60	02/05/10	02/10/10	Patrick Dupont
☐	1	75.00%	45.00%	Design	500		50	09/19/10	10/19/10	Viviane Ghe
☐	1	6.00%	15.00%	Coding	600		200	06/19/10	11/19/10	Alin Mac
☐	2	20.00%	20.00%	Testing	700		18	09/09/10	03/03/11	Maria Bouchard
☐	3	10.00%	11.00%	Maintenance	800		75	09/19/10	10/19/11	Jacques Demers

Figure 4.17 Performance – Activity level.

4.4.2 Basic reports and graphical representations

Reports are often printed and distributed across the organization to various stakeholders, and the majority of tools in support of PMFs obviously include reports. As expected, reports in MultiPERF contain data about indicators and viewpoints, and are generated in a Portable Document Format (PDF) file. Figure 4.18 contains an example of a basic report that can be generated by MultiPERF.

Name	Coeficient %	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	APV%	TPV%	APV%-TPV%
Understandability	10%	35.00%	50.00%	15.00%								14.10%	84.50%	-70.40%
Learnability	15%		25.00%			10.00%		5.00%	35.00%	10.00%	15.00%	75.00%	80.00%	-5.00%
Operability	50%		30.00%	10.00%	5.00%	25.00%				10.00%	20.00%	35.54%	32.82%	2.72%
Availability	25%		70.00%	5.00%	25.00%							90.64%	90.64%	

Figure 4.18 Sample basic report generated by MultiPERF.

Using graphical representations will facilitate the interpretation and understanding of performance. Colors are used for the graphs in this section by randomly assigning them, and they do not have any significance. As shown in Figure 4.26, all the graphs in the MultiPERF prototype are interactive, meaning that each row (Date) and column (Viewpoints) can be hidden (as are the column that contains the quality viewpoint and the rows that contain the date: 01.14.2010 and 03.14.2010) or displayed on demand.

One-dimensional graphs

One way to see the importance of each measure within an indicator is by using a pie chart, and breaking it down into the various slices, as shown in Figure 4.19.

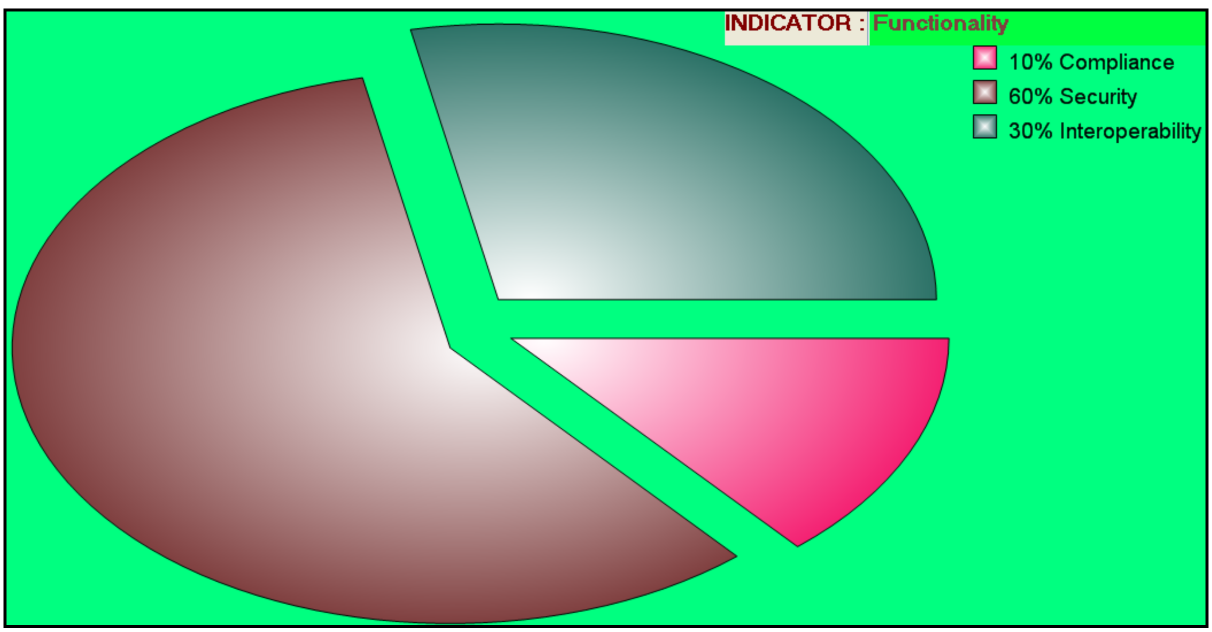


Figure 4.19 Importance of each measure within the Functionality Indicator.

The importance of each measure or indicator within a viewpoint can also be shown in a pie chart, as shown in Figure 4.20, where 3 of 15 indicators don't have any impact on the Economic Viewpoint, and Indicator_20 with 0.3753 has the greatest impact.

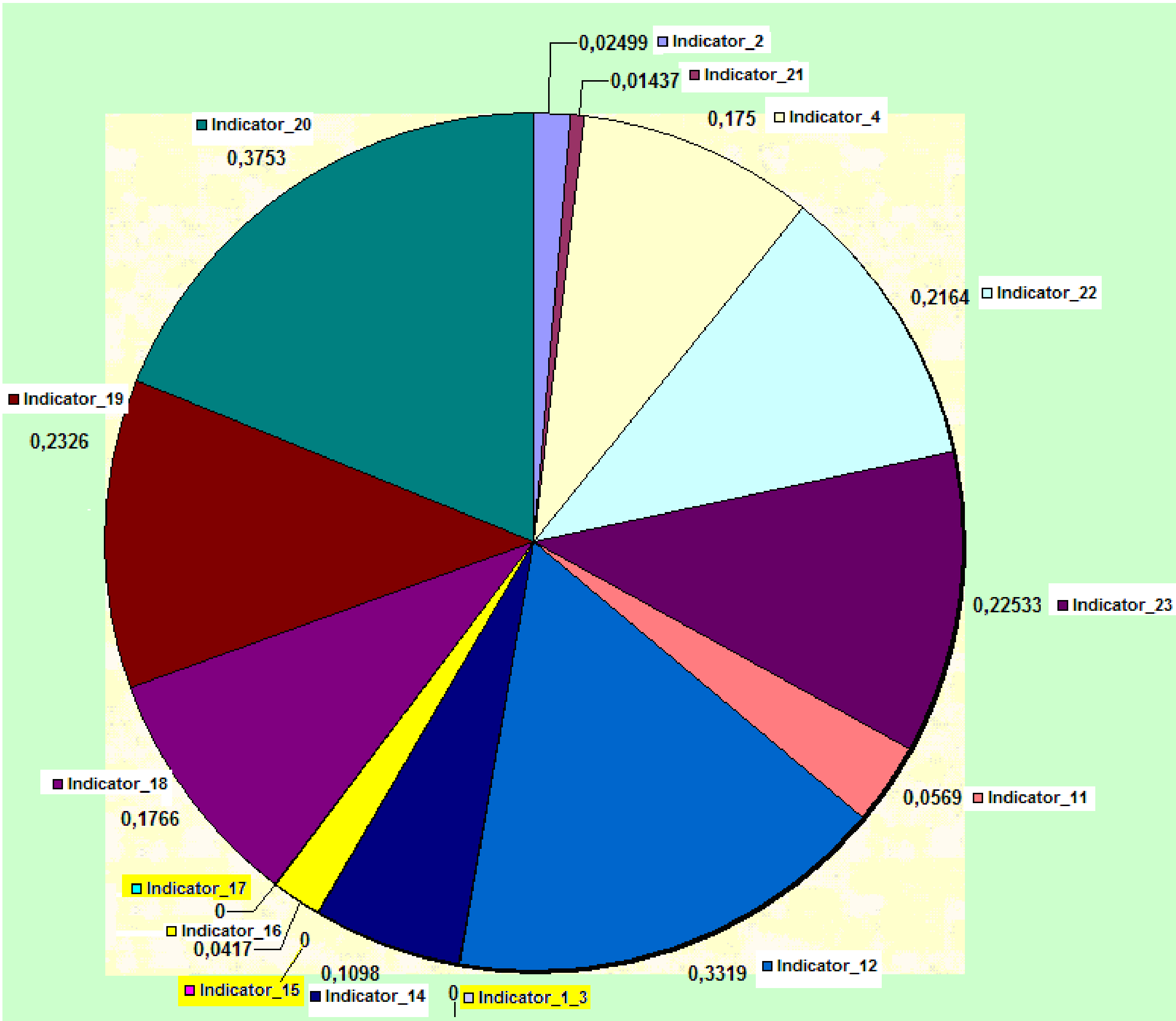


Figure 4.20 Importance of each indicator within the Economic viewpoint.

Two-dimensional graphs

The histogram is a simple graph for visualizing the distribution of the weights of indicators and measures within each viewpoint. When producing this graph, it is possible to select a subset of measures, indicators, and viewpoints, as shown in Figure 4.21. Fourteen indicators and 10 viewpoints have been selected. We can see, for example, that Measure 11 is the most important for the Economic viewpoint, and that Availability is the most important for the Social viewpoint. In addition, we can see that most of the weights are under 40% and that only three viewpoints have indicators greater than 50%: Economic: Compliance and Measure

11; Social: Availability; and Quality of Work Life : Maturity, Fault tolerance, Recoverability, Measure 9, Measure 10, and Measure 12. Also, Measure 11 is assigned only the Economic viewpoint, and Usability is assigned to the most viewpoints: 8.

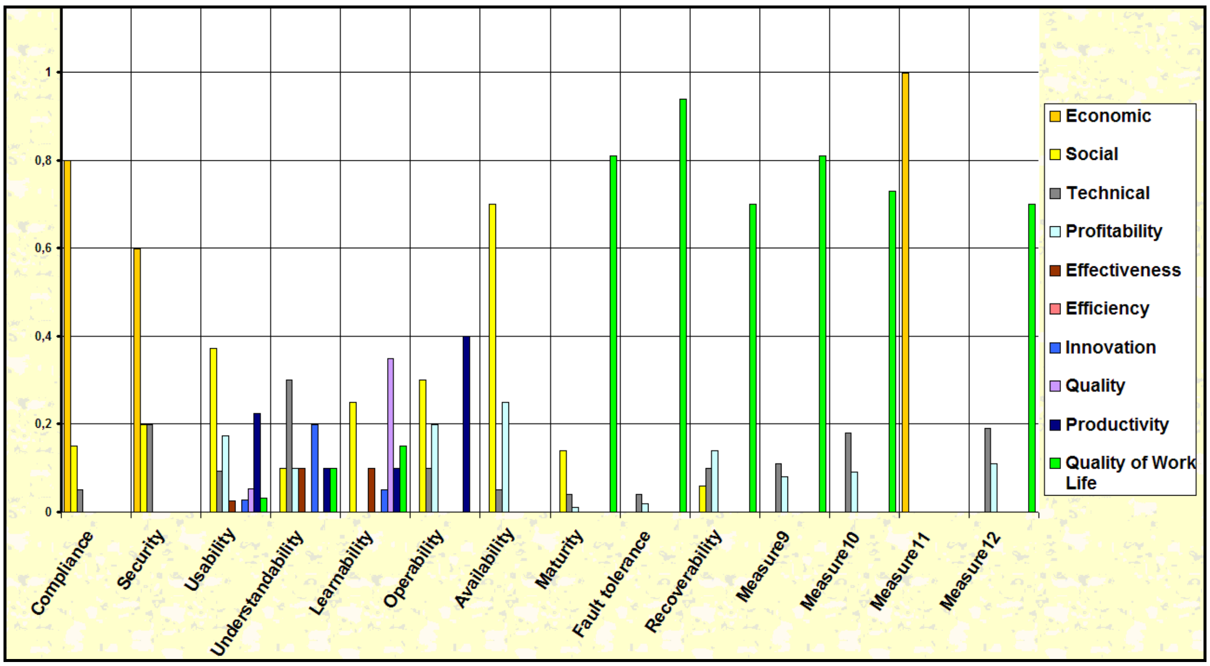


Figure 4.21 Histogram of indicator weights within viewpoints.

The area block graph is easy to understand. An example is shown in Figure 4.22 and Figure 4.23 (histogram graph), where, for example:

- Measure 11 has no impact on any viewpoint other than the Economic one;
- Compliance and Security have a strong impact on the Economic viewpoint, and also on the Technical and Social viewpoints;
- There is not one indicator with the same impact on all viewpoints;
- The Efficiency viewpoint is not selected at all;
- The Quality of Work Life is impacted by the largest number of indicators, which is evident from the large area of green in Figure 4.22;

- Even though the Technical viewpoint is impacted by almost all the indicators, it is still only one-third as important as Quality of Work Life. The situation is similar for the Social viewpoint, which is almost half as important as Quality of Work Life.

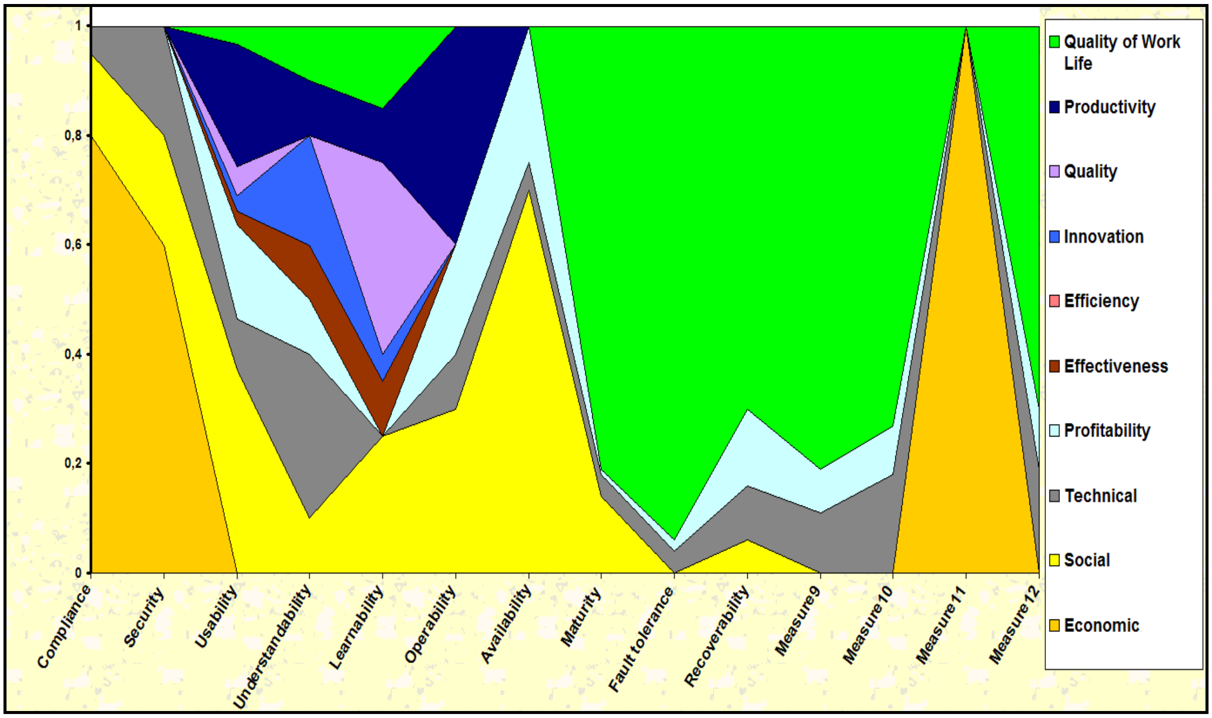


Figure 4.22 Area block of indicator weights within viewpoints.

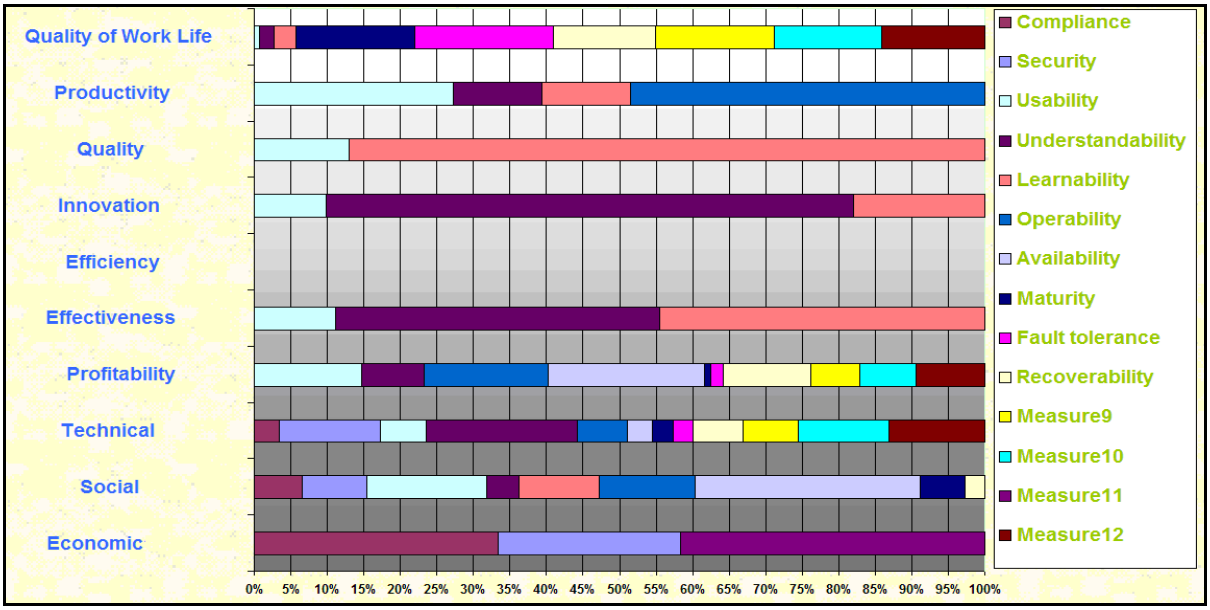


Figure 4.23 Histogram of viewpoints weights within indicators.

Simple graphical representations to identify the gap between the APV% and the TPV% for viewpoints are the line graph and the Kiviat graph, as shown in Figure 4.24 and Figure 4.25, where, for example, the APV% and the TPV% on 04.29.2010 are compared. In these figures, we can see the following:

- Only two viewpoints out of ten greatly exceed the expectations of the organization: Innovation and Profitability;
- The Effectiveness and Economic viewpoints are the ones with unacceptable performance;
- Innovation has the best performance, and Effectiveness the worst performance.

The evolution of performance over time can also be analyzed using a histogram, as shown in Figure 4.26.

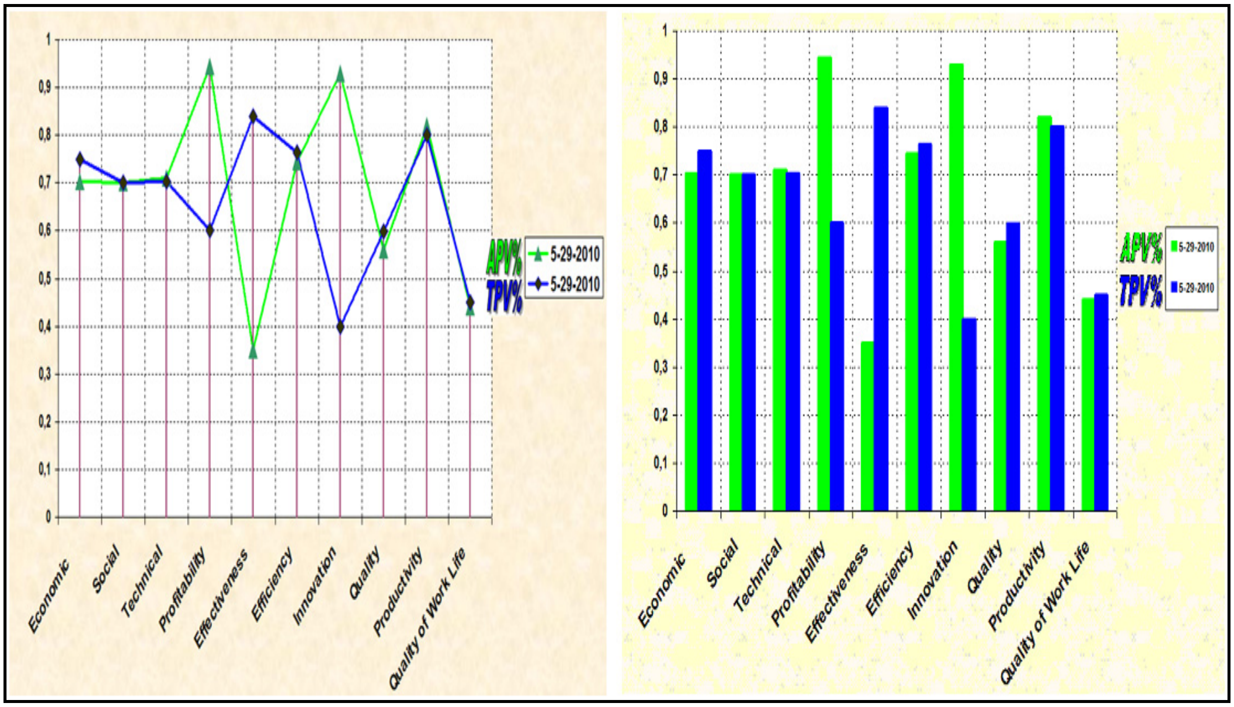


Figure 4.24 Line graph (left side) and histogram (right side) comparing APV% with TPV% for each viewpoint.

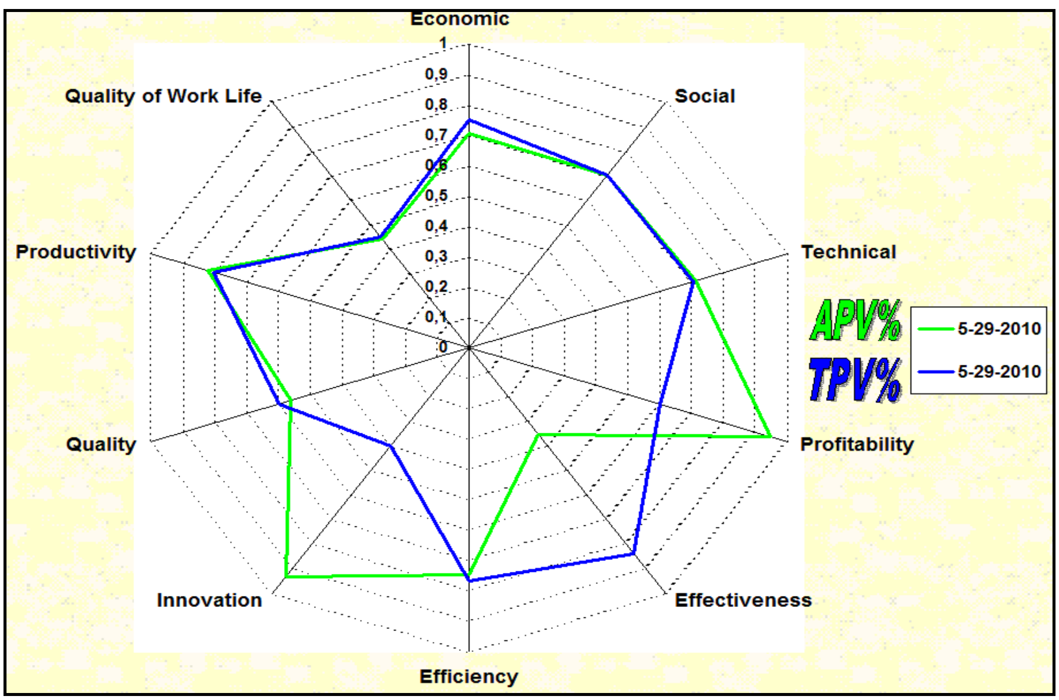
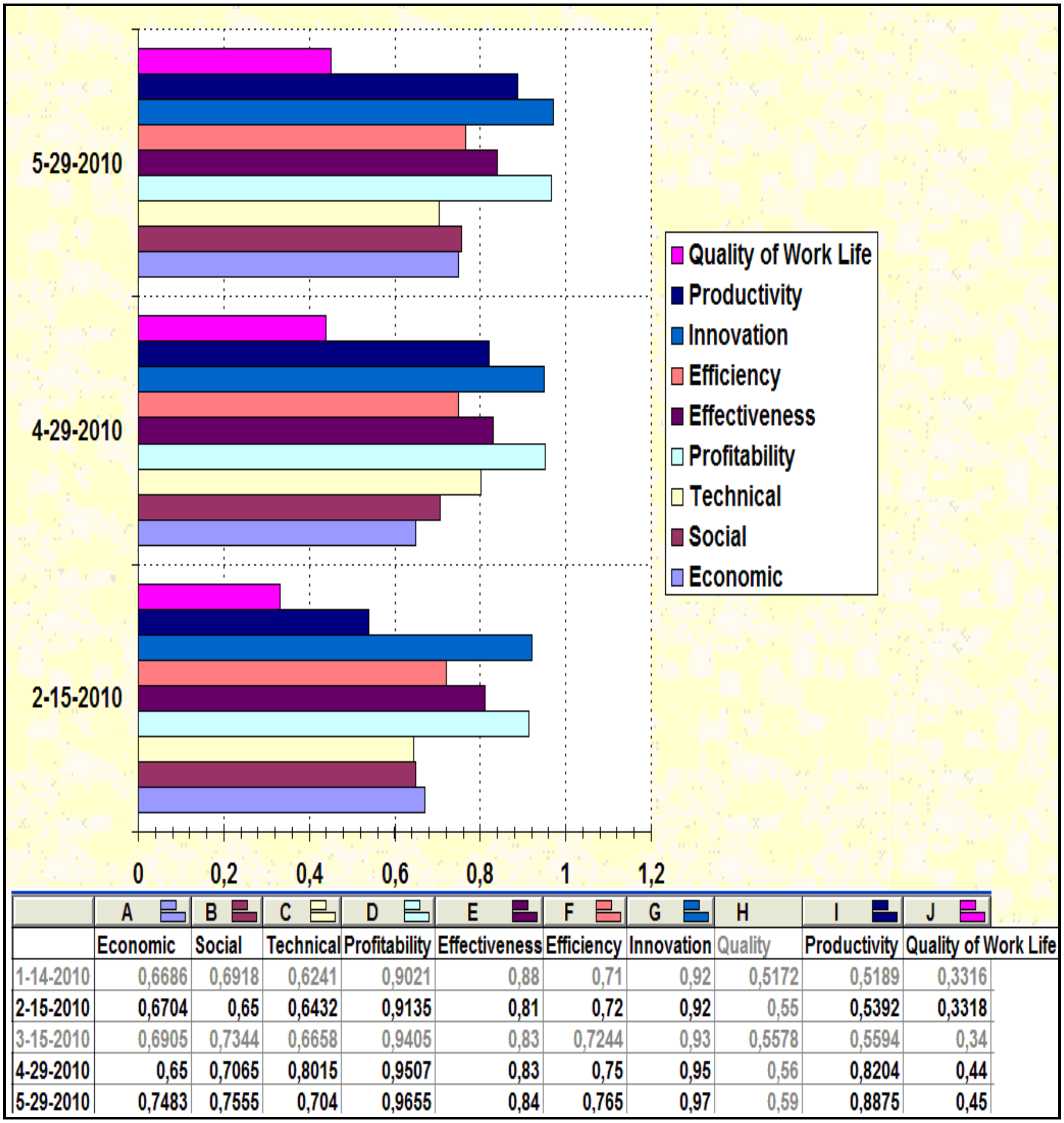


Figure 4.25 Kiviat graph comparing APV% with TPV% for each viewpoint.

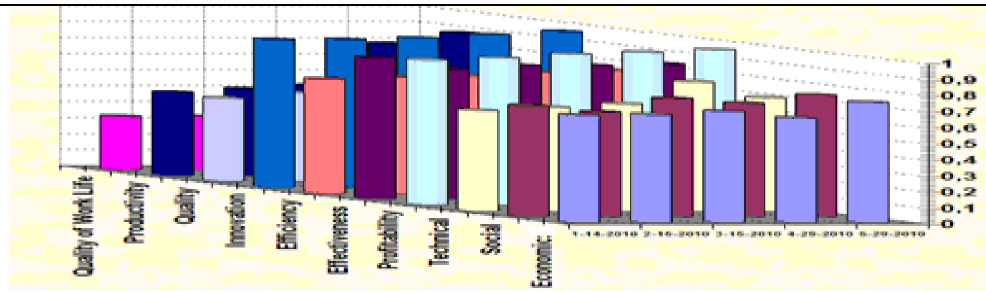
As shown in Figure 4.27, a set of interactive three-dimensional graphs is also available in MultiPERF. In addition, each row (performance date) and column (viewpoints) can be selected, or not, in these graphs.



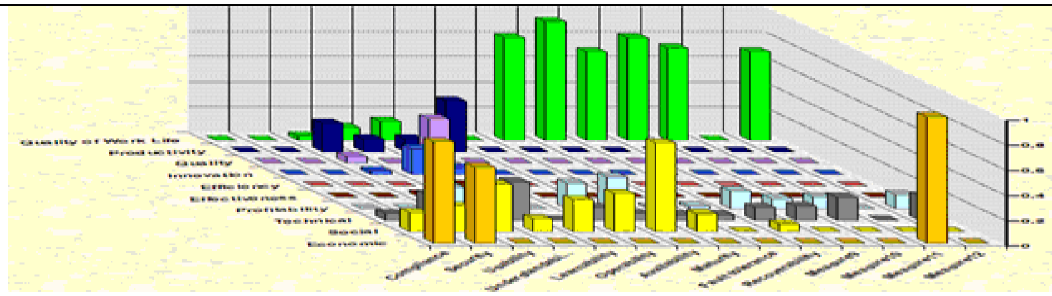
	A	B	C	D	E	F	G	H	I	J
	Economic	Social	Technical	Profitability	Effectiveness	Efficiency	Innovation	Quality	Productivity	Quality of Work Life
1-14-2010	0,6686	0,6918	0,6241	0,9021	0,88	0,71	0,92	0,5172	0,5189	0,3316
2-15-2010	0,6704	0,65	0,6432	0,9135	0,81	0,72	0,92	0,55	0,5392	0,3318
3-15-2010	0,6905	0,7344	0,6658	0,9405	0,83	0,7244	0,93	0,5578	0,5594	0,34
4-29-2010	0,65	0,7065	0,8015	0,9507	0,83	0,75	0,95	0,56	0,8204	0,44
5-29-2010	0,7483	0,7555	0,704	0,9655	0,84	0,765	0,97	0,59	0,8875	0,45

Figure 4.26   Evolution of organizational performance for each viewpoint.

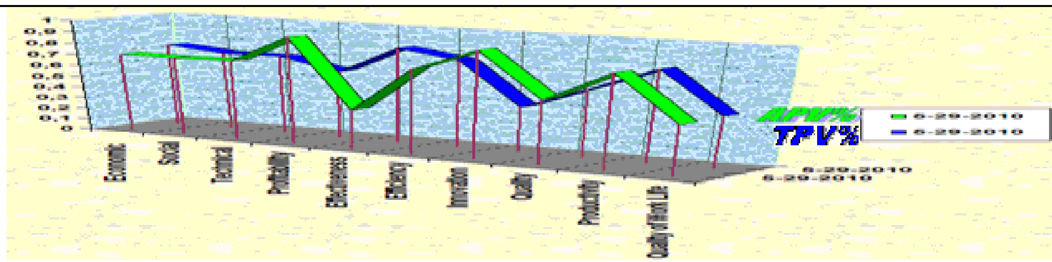
Cylinder



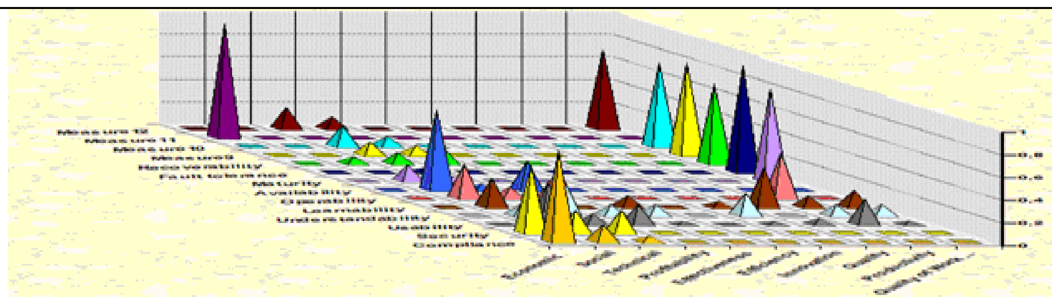
Bar



Curve



Pyramid



Surface

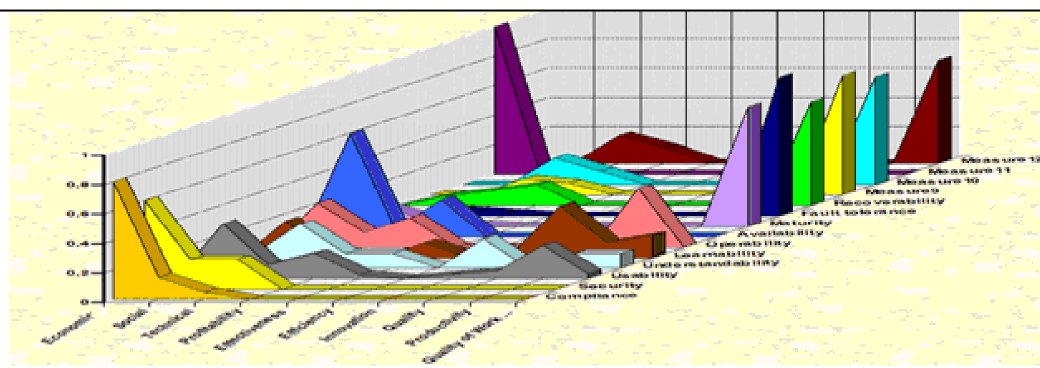


Figure 4.27 Three-dimensional graphs.

4.4.2.1 Other available features

Time is an important dimension to take into consideration when managing performance. Every organization evolves over time. It is important to know the TPV% (planned) and APV% (actual) at different times:

- What are the current APV% and TPV%?
- What were the APV% and TPV% a number of months ago?
- What is the next predicted TPV%?

For visualization purposes, the time axis is divided into two parts in Figure 4.28: the APV% is in the top part, and the TPV% is in the lower part. Comparing the APV% and the TPV% on different dates is useful for managers, as shown in Figure 4.28, where the APV% and TPV% of 31 May, 2010, are compared.

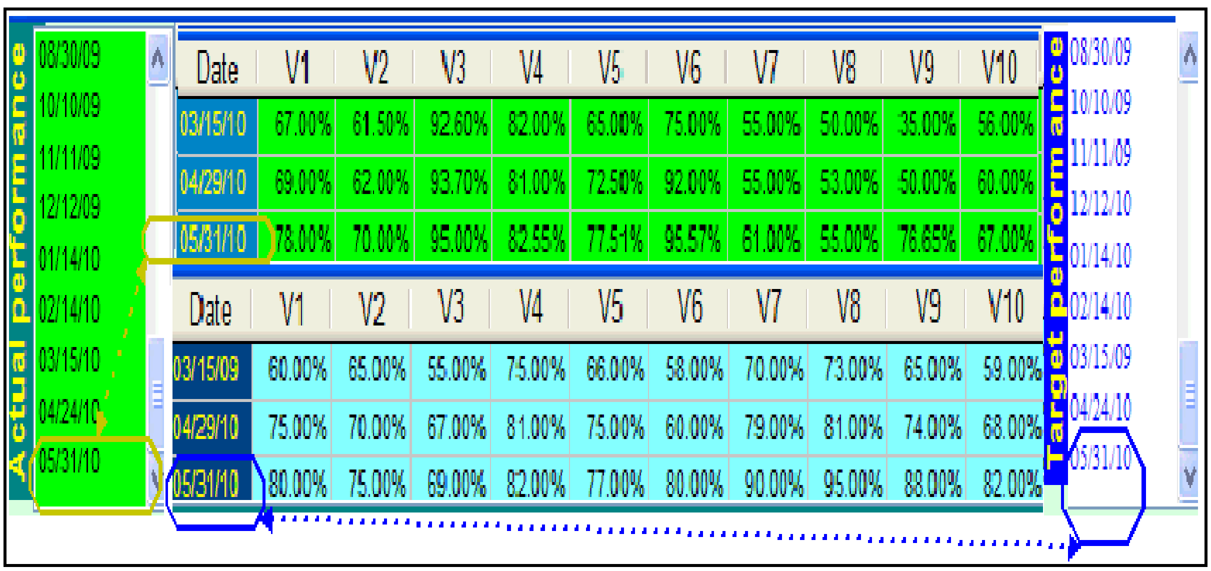


Figure 4.28 Comparing APV% and TPV%.

4.4.3 Building the pyramid

This section shows how to build a graph, a pyramid in this case, to represent the many possible performance viewpoints graphically and in a consolidated manner, while at the same time keeping track of the values of the individual indicators and measures.

Polygon – Polyhedron – Pyramid

A **polygon** is 'the union of a set of coplanar segments, each of which intersects just two other segments, neither one collinear with it, one at each end point' (Jurgensen *et al.*, 1965, p. 61), a triangle being the simplest polygon: "A triangle consists of three non collinear points. The segments are sides and the points the vertices (plural of vertex) of the triangle." (Smith *et al.*, 1992, p. 149). A vertex represents a point where two or more straight lines meet. **Polyhedrons** are "space figures…which are made up of only polygonal regions… The polygonal regions are the faces of the polyhedron. We call the sides and vertices of the faces edges and vertices of the polyhedron, respectively" (Smith *et al.*, 1992, p. 485)

A **pyramid** "is a polyhedron one of whose faces is a polygon, and whose other faces are triangles having a common vertex and the sides of the polygon for bases" (Weisstein, 2002, p. 302) and, depending on the shape of the polygon at the base, a pyramid can be correspondingly triangular, quadrangular, pentagonal, hexagonal, and so on. A triangular pyramid, or tetrahedron, has a triangle as its base. An irregular pyramid or non regular pyramid has unequal lateral edges, or the base is not a regular polygon. Otherwise, it is a regular pyramid.

The geometrical figure that has been chosen to represent performance is a regular pyramid, as shown in Figure 4.29**.** The pyramid's base vertices represent the lowest level of performance for each viewpoint: the performance values are normalized, and therefore the pyramid base vertices are always equal to zero. The top vertex of the pyramid – the convergence of all the vertices (the apex) – corresponds to the top level of performance for each viewpoint, and will always be equal to one. If performance is managed as a project, the pyramid's base vertices represent the beginning of the project and the apex represents the end of the project.

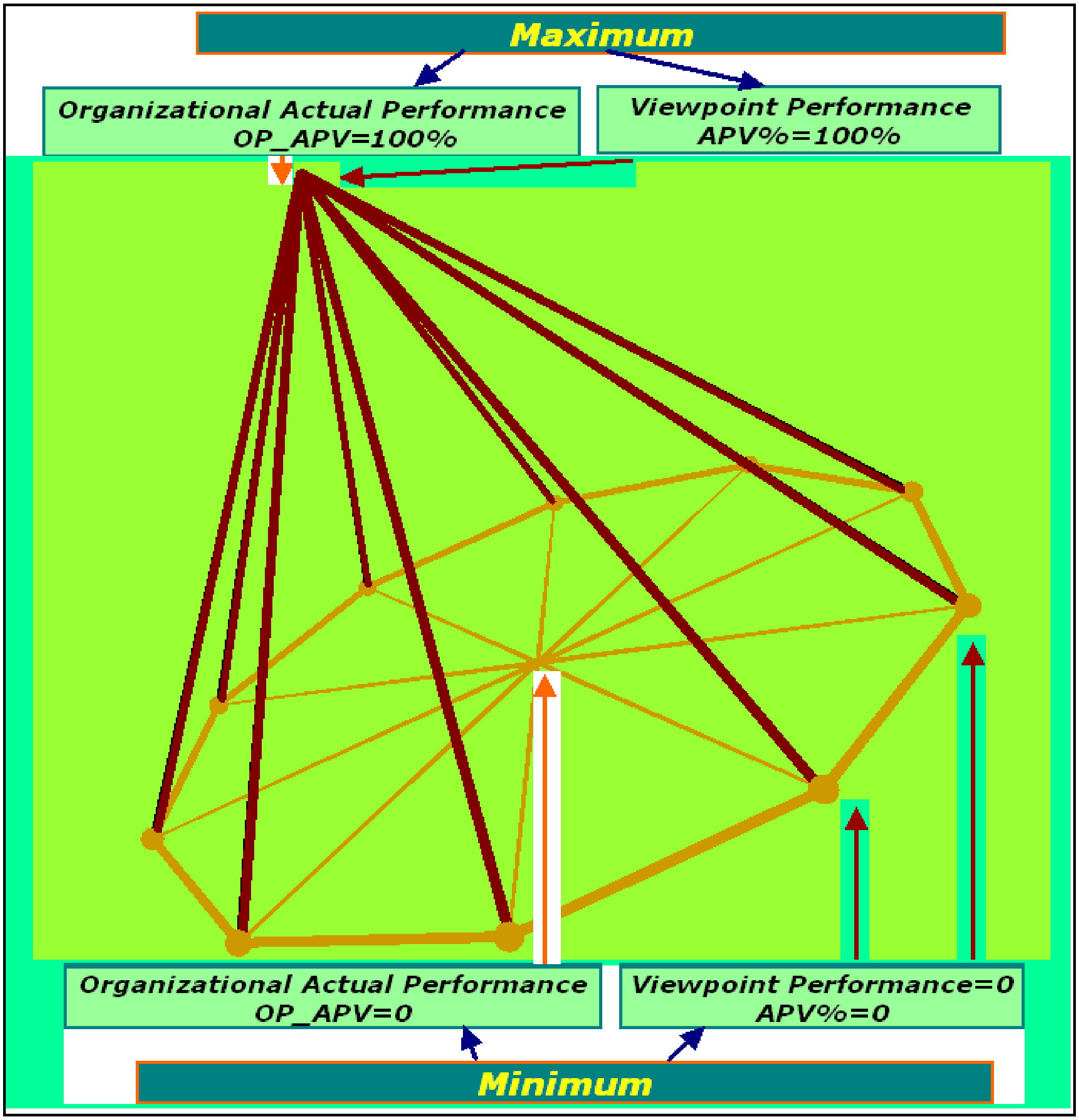


Figure 4.29 Organizational performance using a pyramidal representation.

4.4.3.1 Using a sphere to represent the APV% for a viewpoint

The geometrical figure chosen to represent the APV% of a viewpoint is a sphere: "Perhaps the most common geometric solid of all is the sphere… A sphere is the locus of all points in space a given distance from a given point." (Smith *et al.*, 1992, p. 511).

A sphere represents the APV%, as shown in Figure 4.30. There are two possibilities available in the prototype: using a sphere with the same radius for all viewpoints, as shown in Figure 4.30 (A), or using a sphere with a radius proportional to the weight of the viewpoint, as shown in Figure 4.30 (B).

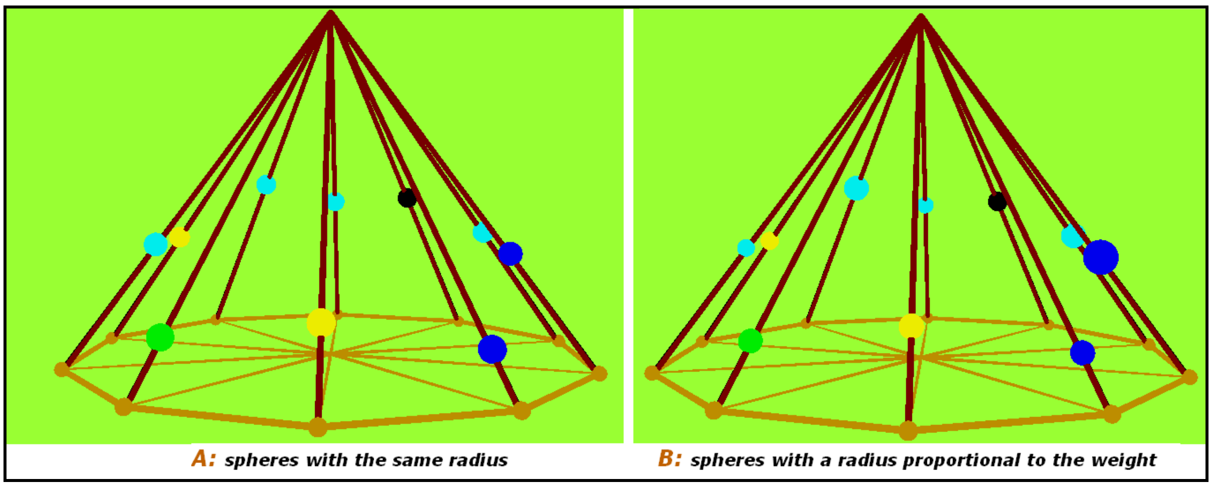


Figure 4.30 Representation of the APV% using a sphere for each viewpoint.

The APV% of each viewpoint are classified in three levels, as shown in Figure 4.30, and the colors of the sphere have the following meanings, as shown in Table 4.3.

Table 4.3 Color legend of spheres

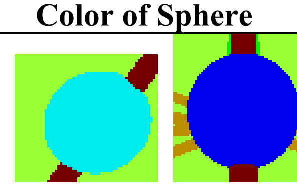
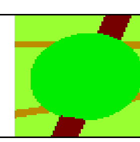
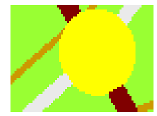
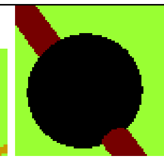
Performance	**Interpretation**	**Color of Sphere**
Excellent	A blue sphere means that the performance for this viewpoint exceeds the expectations of the organization. Dark blue means that the APV% is greater than the TPV% by more than 10%, and the performance is therefore considered outstanding.	
Acceptable	A green sphere means that the performance for this viewpoint meets all expectations.	

Table 4.3 Color legend of spheres (continued)

Unacceptable	A black or yellow sphere means that the performance for this viewpoint fails to meet expectations. Yellow means that the TPV% is greater than the APV% by less than 5%, and black means that this difference is more than 5%. Why was black chosen and not red? Simply because a black sphere shows up better than a red one on a red surface.	 

4.4.3.2 Using a cube to represent the OP_APV and OP_TPV of a viewpoint

A projection of OP_APV and OP_TPV using a cube on every viewpoint is shown in Figure 4.31, where target performance is higher than actual performance:

- A dark magenta cube represents the target performance;
- A light olive green cube represents the actual performance;
- A green cube represents target performance and actual performance when OP_APV=OP_TPV.

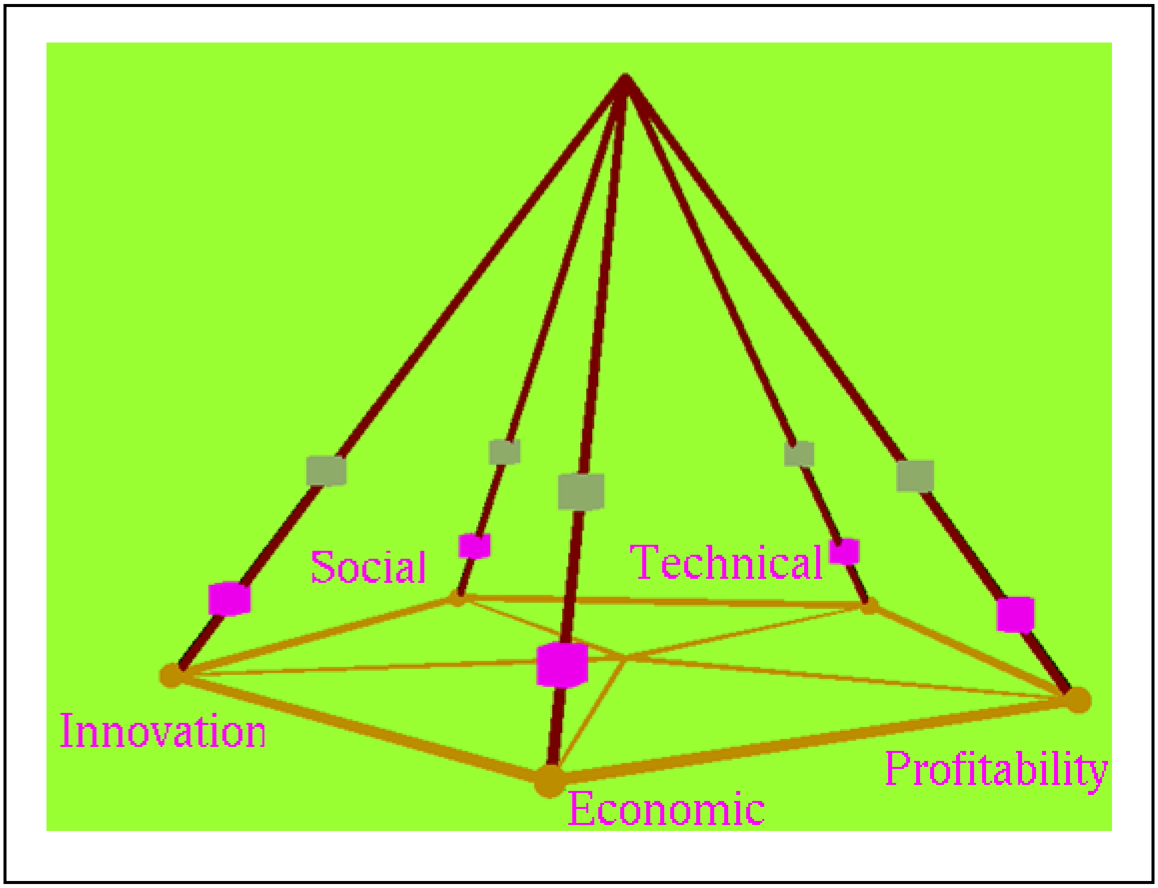


Figure 4.31 OP_APV and OP_TPV represented using a cube.

4.4.3.3 Using a line to represent TPV%

A projection of TPV% using a line on every viewpoint is shown in Figure 4.32, where the next three performance targets are represented. The previous TPV% is not shown in the graph. The next TPV% are represented using an aqua blue line. There is no limit to the number of TPV% that can be shown. A maximum volume of the upper part means that this is the start of the project, as is shown in Figure 4.32.

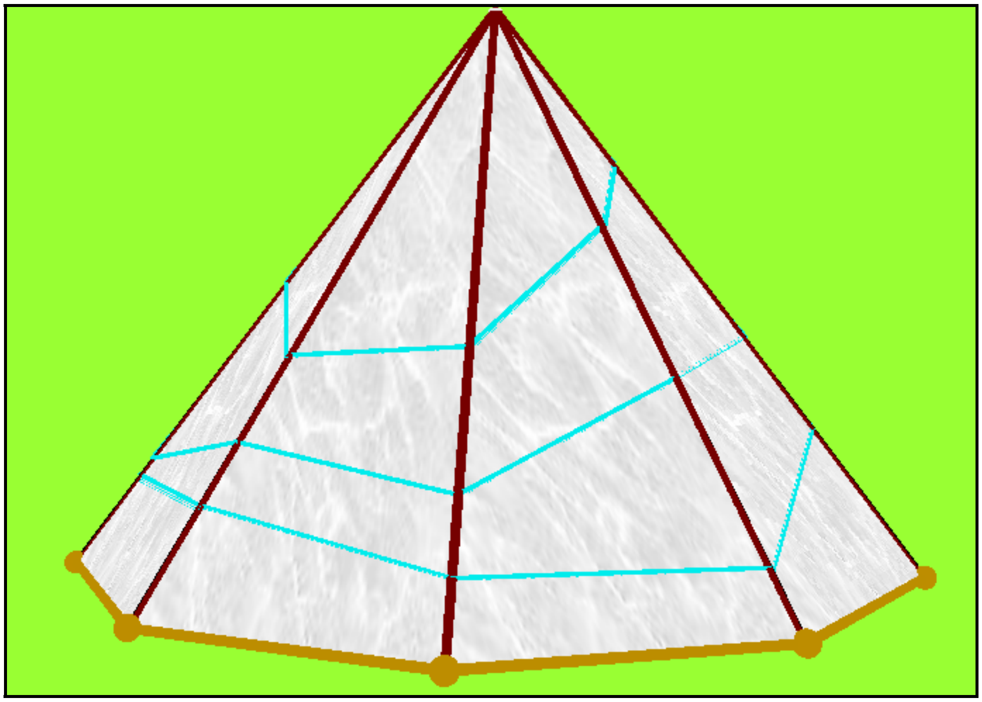


Figure 4.32 Performance – an aqua line indicating the next TPV%.

4.4.3.4 OP_APV and OP_TPV are indicated by a cone

A cone is a space figure with a vertex and a circular base, as shown in Figure 4.33 and as defined in (Smith *et al.*, 1992, p. 510): "The circular region is called the base of the cone. The region joining the vertex and the base of a cone is the lateral surface. The height of a cone is the length of the perpendicular from the vertex to the plane of the base."

Cones are used because the apex of the cone is easy to visualize in the prototype, especially when the pyramid is rotated. The volume of the cone has no significance. The axis of the cone is the straight line from the center of the pyramid passing through the apex, as shown in Figure 4.33.

Starting from the bottom of the pyramid, a cone in bright green represents the OP_APV. If the OP_APV is lower or greater than the OP_TPV, the cone will be cut off by a plane parallel to its base, resulting in a truncated cone or frustum, as shown in Figure 4.33. If the

OP_APV is greater than the OP_TPV, this difference (this part of the cone) will be colored in blue, as shown in Figure 4.33. If the OP_APV is lower than the OP_TPV, this difference will be colored in red. From the top or apex of the pyramid, another cone in the opposite direction colored in white represents the difference between the maximum point represented either by OP_APV or by OP_TPV and the apex of the pyramid, as shown in Figure 4.33. If performance is managed as a project, the center situated at the bottom of the pyramid and represented by the green cone is the start of the project and the apex is the end of the project.

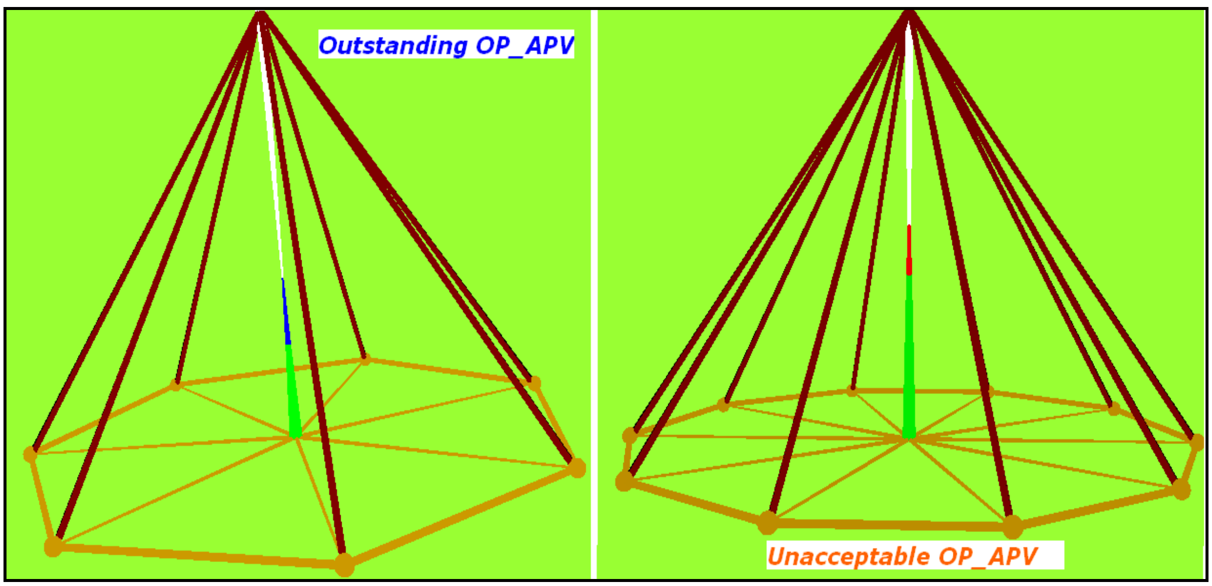


Figure 4.33 OP_APV : Outstanding (left), and unacceptable (right).

4.4.3.5 Using a line to represent APV% and TPV%

A white line links the TPV% values, as shown in Figure 4.34 and Figure 4.35, and a grey line links the APV% values, as shown in Figure 4.35, where the grey line is consistently higher than the white line, because, at every viewpoint, the APV% is higher than or equal to the TPV%. This is obviously not always the case: the grey line can, of course, be lower than the white line.

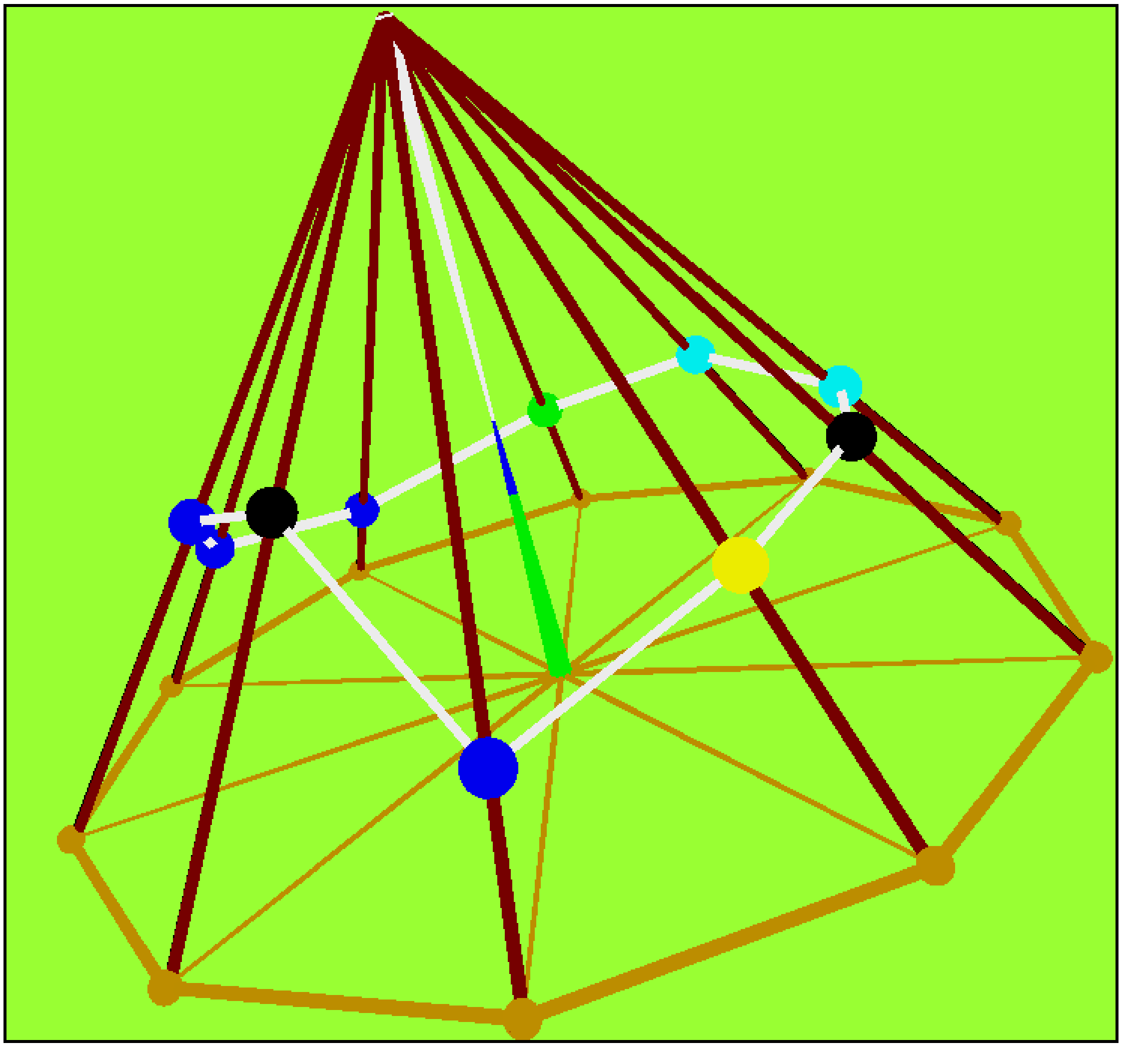


Figure 4.34 Performance – white line delimiting the TPV.

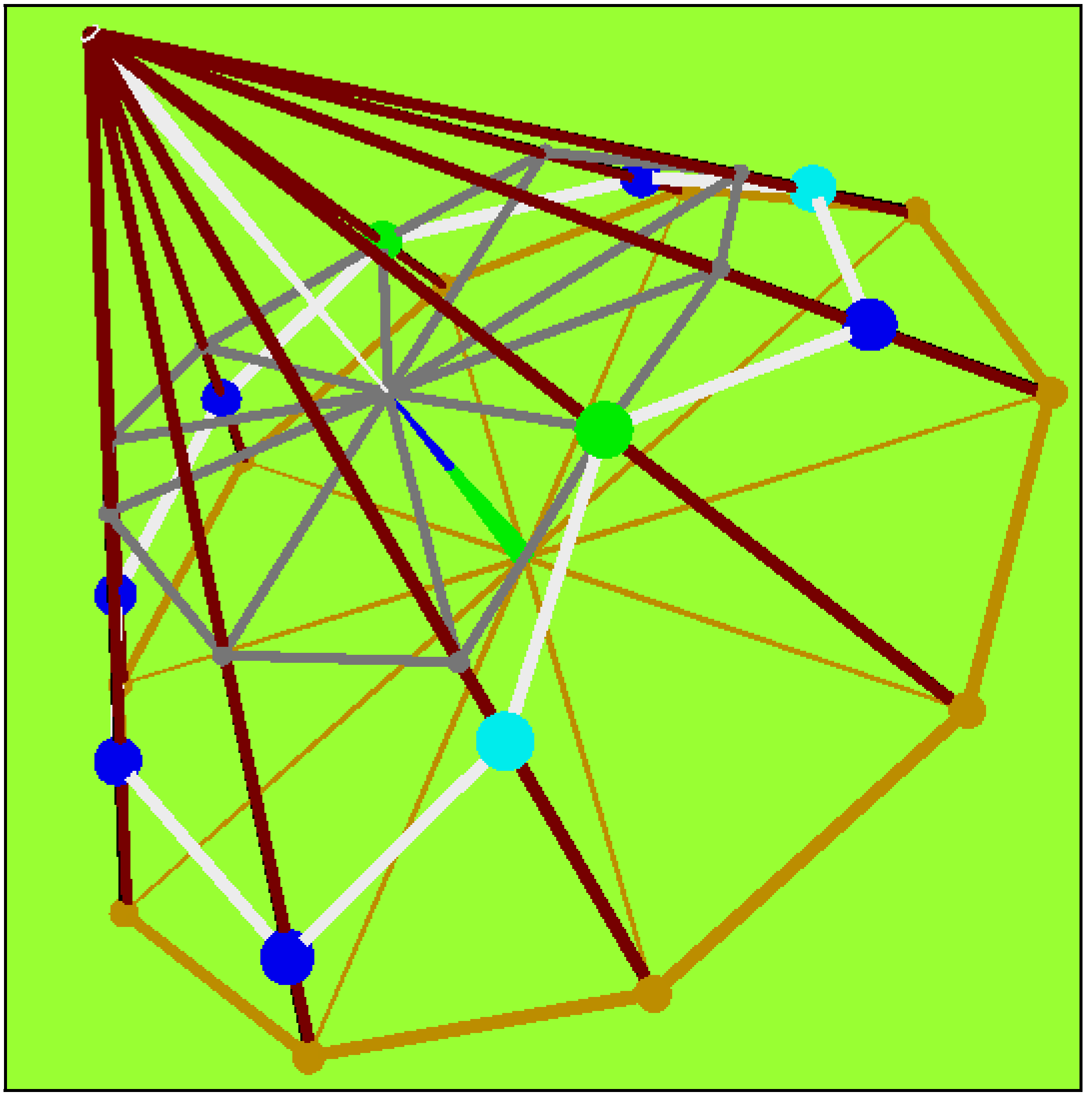


Figure 4.35 Line delimiting the APV (grey line) and TPV (white line).

4.4.3.6 Volume concept to represent organizational performance

The OP_TPV and OP_APV can also be represented using the concept of volume, as shown in Figure 4.36, where the viewpoints of performance in the graph correspond to the pyramid's lateral edge: the minimum performance is situated at the base vertices and the maximum performance is at the top of the pyramid (apex).

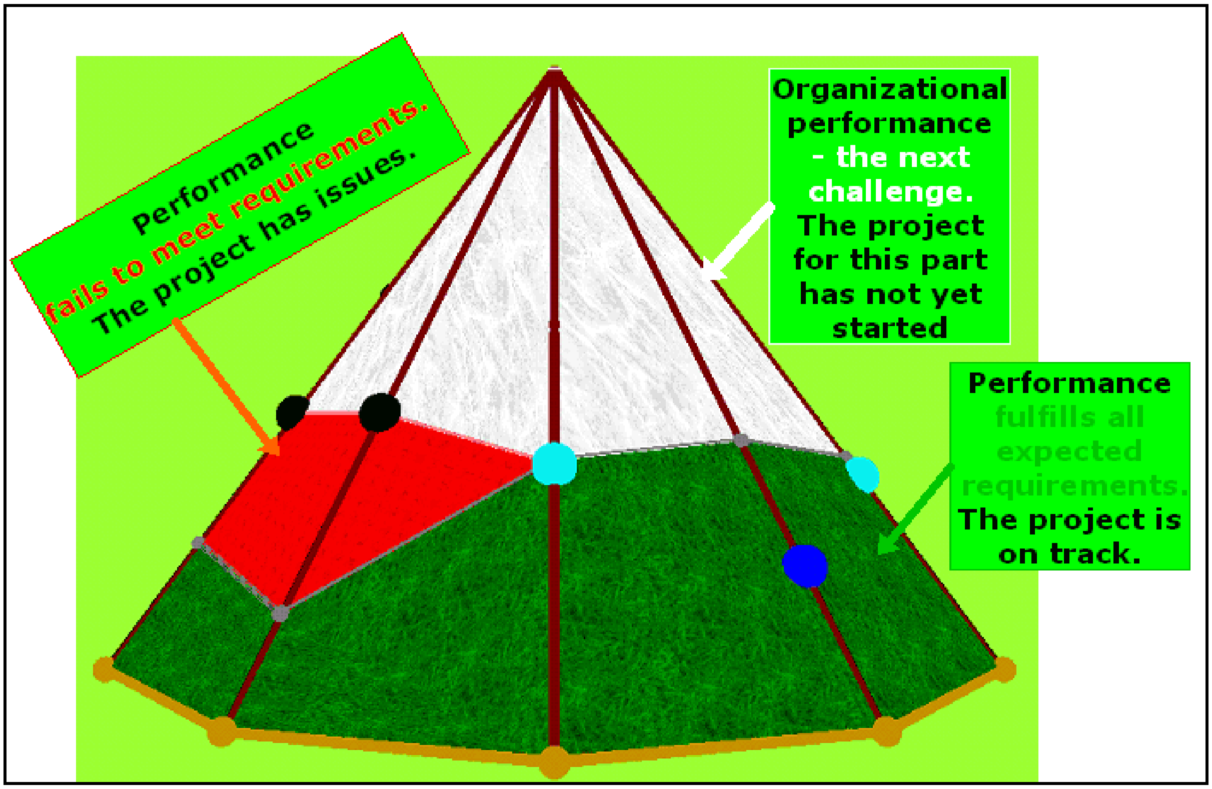


Figure 4.36 How to interpret the color scheme of volumes in the pyramid.

The lines linking the TPV% and APV% of the viewpoints will form different tetrahedrons inside the pyramid. As shown in Figure 4.36, three colors are used to represent performance using the concept of volume: the upper part, the middle part, and the lower part.

Upper part

The volume of the upper part as is shown in Figure 4.36, will be at its maximum at the start of the project (APV% = 0%, since the project has not yet started) and will be at its minimum at the end of the project. (APV% = 100% for every viewpoint).

It is delimited by the higher coordinates between APV% or TPV% for every viewpoint, the higher coordinates between OP_APV or OP_TPV, and the apex of the pyramid. The upper part of the pyramid is colored white, and represents the next challenge for organization. When using the project to manage the performance of a project, this is the part of the project that is not yet completed.

Middle part

The red in the middle part of the pyramid indicates that performance fails to meet expectations for these viewpoints. The negative difference between APV% and TPV% of a viewpoint is represented in red.

The red volumes shown in Figure 4.36 and Figure 4.37 indicate that the APV% fails to meet the TPV% for this viewpoint:

- The black and yellow sphere shows automatically that the performance of this viewpoint is unacceptable.
- The volume between TPV%, OP_TPV, APV%, and OP_APV will be in red for these viewpoints.

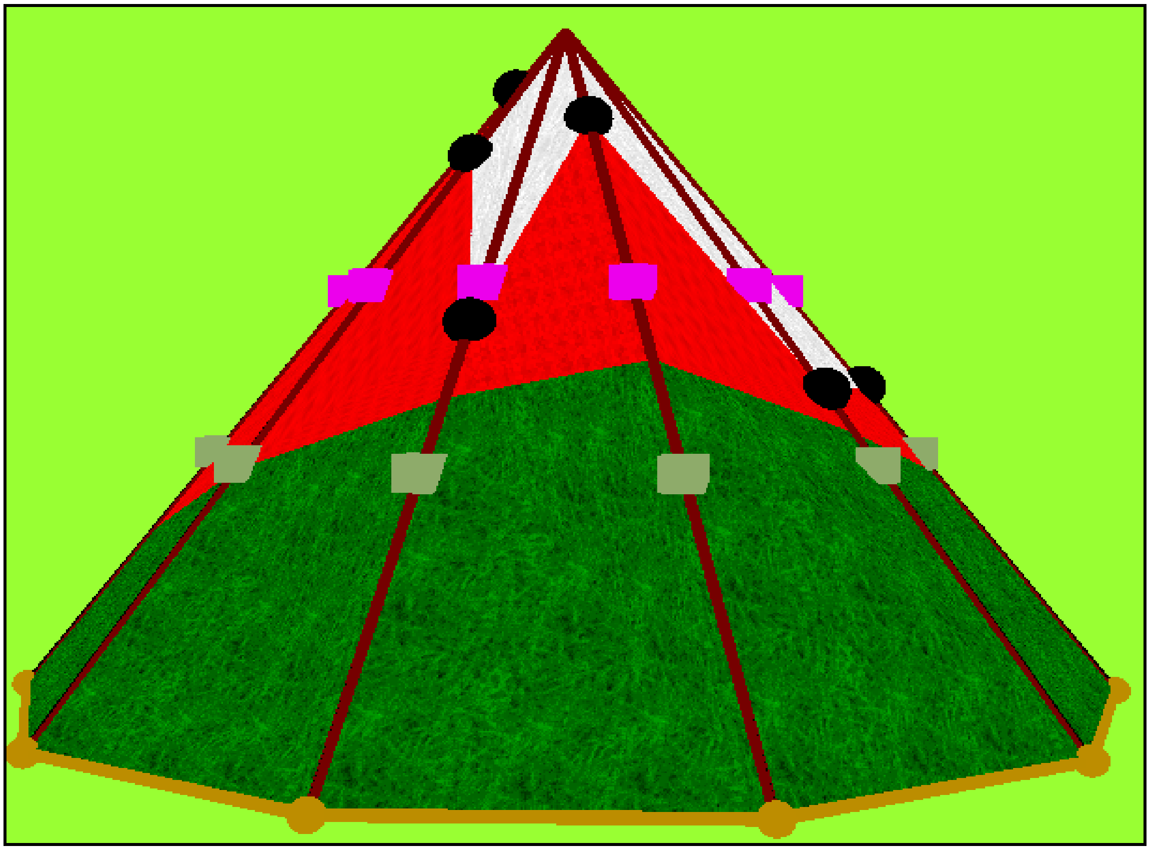


Figure 4.37 Unacceptable OP_APV: the spheres are black and the volume is in red.

Lower part

- Green in the lower part of the pyramid as is shown in Figure 4.36 and Figure 4.38 means that performance fulfills all the expected requirements for this part, and that the project is on track and under control. This area is delimited by the coordinates between the APV% for every viewpoint, the base of the pyramid, and the coordinates of OP_APV.

How to calculate performance using the geometrical concept of volume

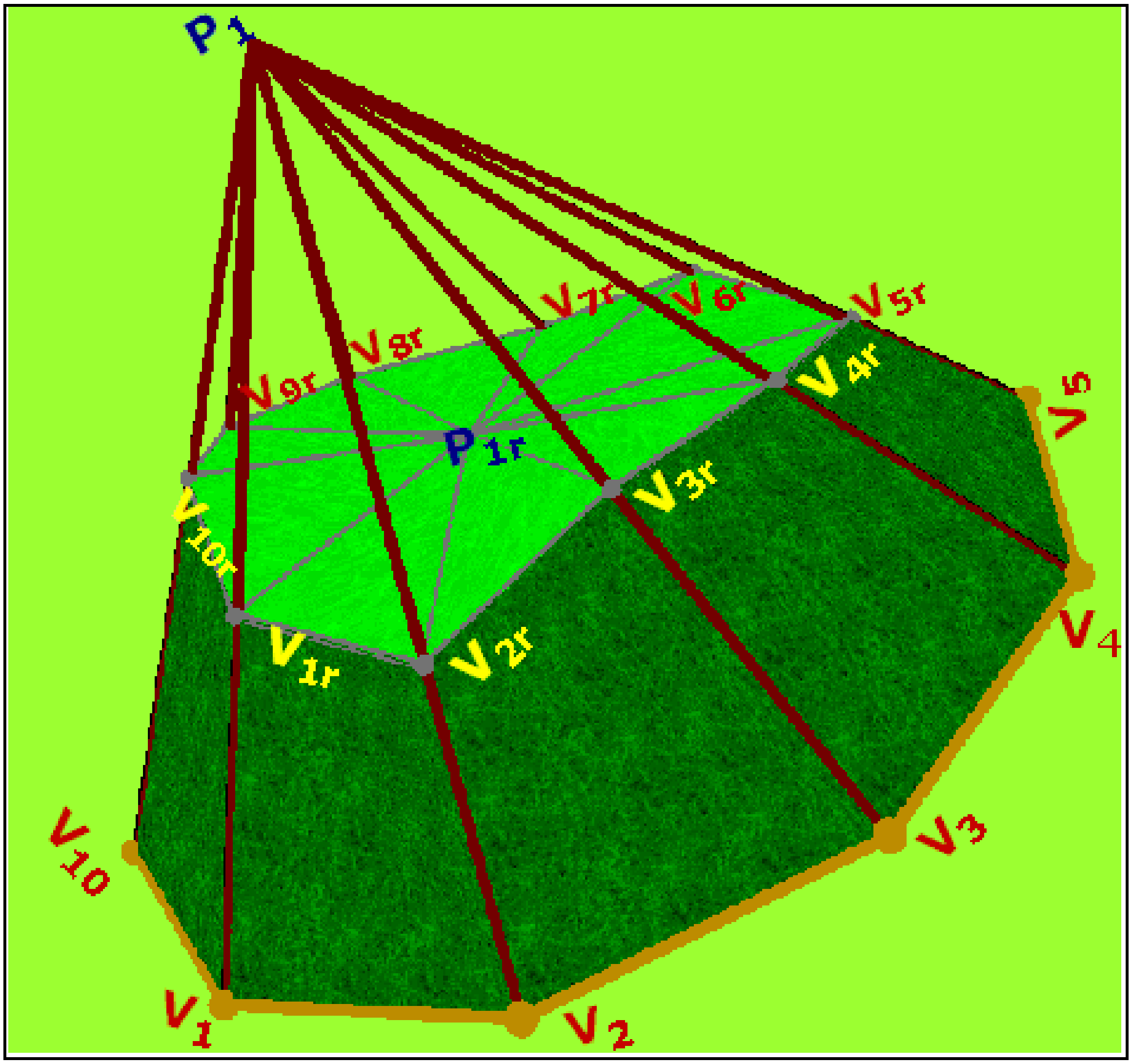


Figure 4.38 Volume concept – Example using ten viewpoints.

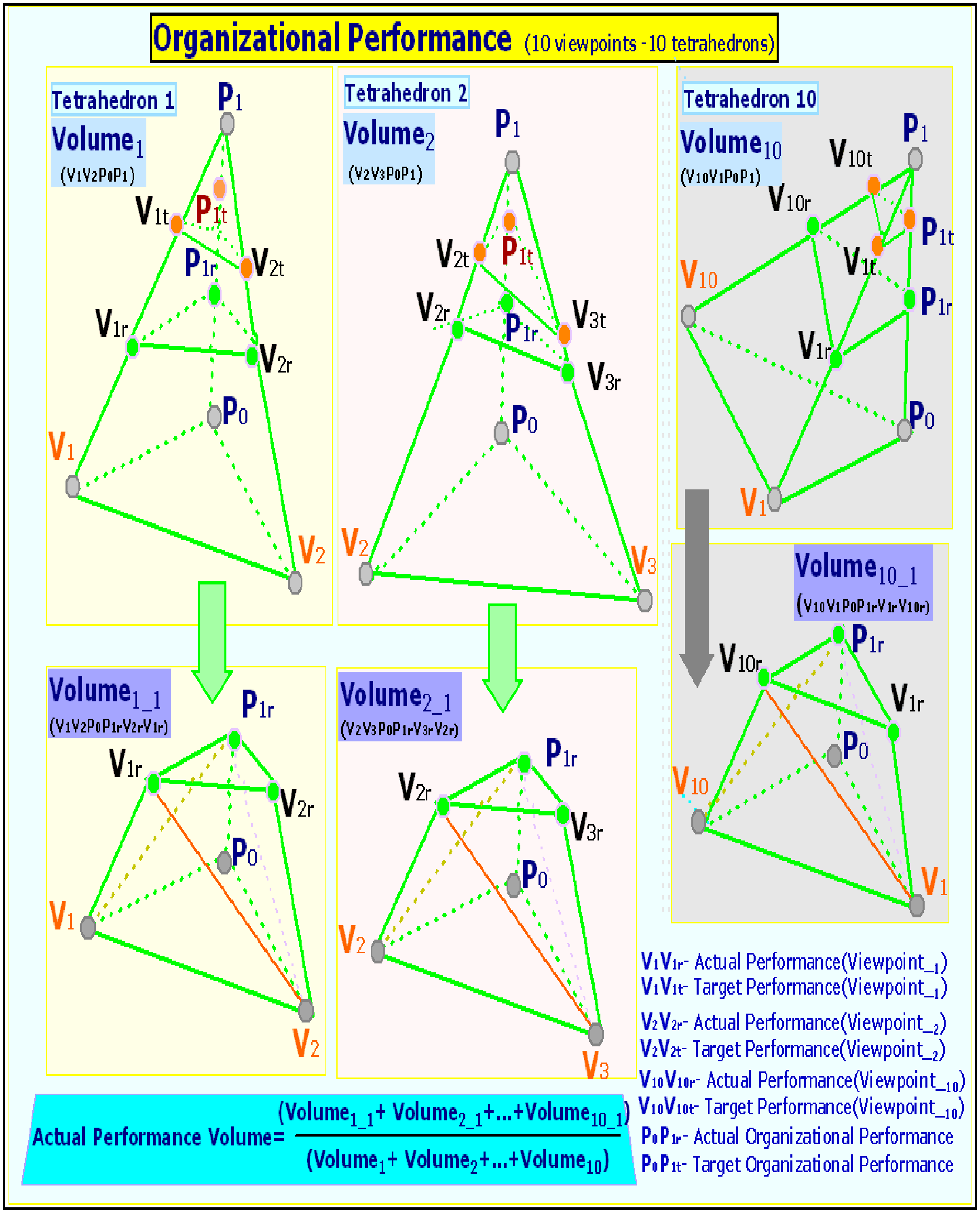


Figure 4.39 Volume concept – Example using ten viewpoints – ten tetrahedrons.

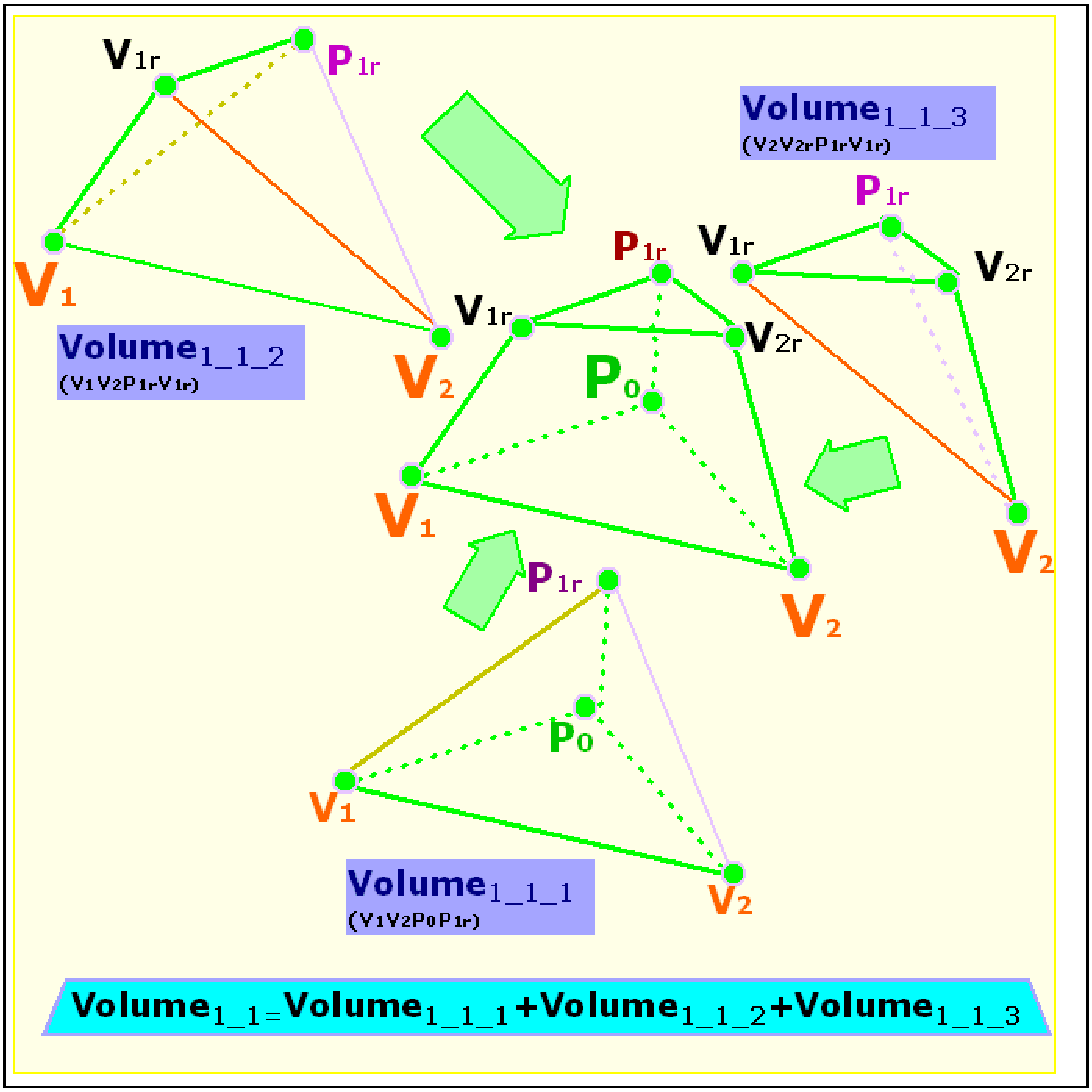


Figure 4.40 Actual performance for tetrahedron 1, as shown in Figure 4.39.

As shown in Figure 4.39, in the construction of the organizational pyramid:

- The triangle is always present, for example: $V_1V_2P_1$, $V_2V_3P_1$, $V_{1r}V_{2r}P_1$, $V_{2r}V_{3r}P_1$, $P_1P_0V_2$;
- The common vertex (the apex) in Figure 4.39 is P_1, and represents 100% of the organizational performance: the lateral faces are $P_1V_1V_2$, $P_1V_1P_0$, and $P_1V_2P_0$, and the base is $P_0V_2V_1$. The altitude of the pyramid (the line between the apex (P_1) and the vertex

(P_0) situated at the base of the pyramid) represents the organizational performance from 0% to 100%, as shown in Figure 4.33, Figure 4.34, and Figure 4.35.

The pyramid is divided into a number of tetrahedrons equal to the number of viewpoints, as shown in Figure 4.38 using 6 viewpoints, or in Figure 4.39 using 10 viewpoints. The vertices (V_1, V_2,…, V_{10}) correspond to the pyramid's base vertices for each viewpoint. The vertices (V_{1r}, V_{2r},…, V_{10r}) correspond to the APV% for every viewpoint, and the vertices (V_{1t}, V_{2t},…, V_{10t}) correspond to the TPV%. The organizational performance vertices are P_0 (bottom performance - performance value = 0), P_1 (top performance - performance value = 1), P_{1r} (Actual performance of Viewpoint_1), and P_{1t} (Target performance of Viewpoint_1).

The volume for $Tetrahedron_1$ ($Volume_1$) will be calculated using the following vertices: $P_1V_1V_2P_0$, and the same operation will be performed for $Tetrahedron_2$ to $Tetrahedron_{10}$, as shown in Figure 4.39. As shown in Figure 4.38, the total volume of the organizational pyramid, Volume (V), is calculated as follows:

$$V = 1 = 100\% = Volume_1 + Volume_2 + ... + Volume_{10}$$

The lower part of pyramid is divided into 30 tetrahedrons, as shown in Figure 4.39:

- $Volume_{1\_1}$ in three irregular tetrahedrons: $Volume_{1\_1\_1}$, $Volume_{1\_1\_2}$, and $Volume_{1\_1\_3}$, as shown in Figure 4.40;
- $Volume_{2\_1}$ in three irregular tetrahedrons: $Volume_{2\_1\_1}$, $Volume_{2\_1\_2}$, and $Volume_{2\_1\_3}$;
- $Volume_{10\_1}$ in three irregular tetrahedrons: $Volume_{10\_1\_1}$, $Volume_{10\_1\_2}$, and $Volume_{10\_1\_3}$.

Finally, the Lower Volume (LV) is shown in green in Figure 4.38, and, as shown in Figure 4.36, and Figure 4.37 and can be calculated as follows:

$$LV = \frac{Volume_{1\_1} + Volume_{2\_1} + ... + Volume_{10\_1}}{Volume_1 + Volume_2 + ...Volume_{10}} \% =$$

$$= \frac{Volume_{1\_1\_1} + Volume_{1\_1\_2} + Volume_{1\_1\_3}}{V}\% +$$

$$+ \frac{Volume_{2\_1\_1} + Volume_{2\_1\_2} + Volume_{2\_1\_3}}{V}\% + \ldots +$$

$$+ \frac{Volume_{10\_1\_1} + Volume_{10\_1\_2} + Volume_{10\_1\_3}}{V}\%$$

The upper part also comprises 10 tetrahedrons, each of which will have a common vertex P_1 (the apex of the pyramid). The other vertices will be the maximum (P_{1r}, P_{1t}), and, for each edge maximum (V_{1r},V_{1t}) to maximum (V_{10r},V_{10t}). The upper part of the tetrahedron, the Upper Volume (UV), can be calculated as follows:

$$UV = \frac{(Upper\ (Volume_1) + Upper\ (Volume_2) + \ldots Upper\ (Volume_{10}))}{V}\%$$

The volume where the performance is unacceptable, the Middle Volume (MV), is calculated as follows:

$$\mathrm{MV = V - UV - LV = (1 - (UV + LV))\%}$$

How the volume of the tetrahedron is calculated in the prototype

According to Weisstein, the volume of any tetrahedron that is not necessarily regular (Weisstein, 2002, p. 2970) can be calculated using a determinant and the dimensional coordinates.

What is a determinant? It is a number associated with each square matrix, and, in the case of a 2x2 and a 3x3 matrix, is (Weisstein, 2002, p. 711) as follows:

$$Det\begin{pmatrix} x_1 & y_1 \\ x_2 & y_2 \end{pmatrix} = x_1 * y_2 - y_1 * x_2$$

$$Det\begin{pmatrix} x1 & y1 & z1 \\ x2 & y2 & z2 \\ x3 & y3 & z3 \end{pmatrix} = \text{x1}*Det\begin{pmatrix} y2 & z2 \\ y3 & z3 \end{pmatrix} \text{ -y1}*Det\begin{pmatrix} x2 & z2 \\ x3 & z3 \end{pmatrix} \text{ +z1}*Det\begin{pmatrix} x2 & y2 \\ x3 & y3 \end{pmatrix} =$$

$$= x_1 * (y_2 * z_3 - z_2 * y_3) - y_1 * (x_2 * z_3 - z_2 * x_3) + z_1 * (x_2 * y_3 - y_2 * x_3)$$

The volume of any tetrahedron bounded by four vertices that are used to identify dimensional coordinates (x_i, y_i, z_i), where i = 1, 2, 3, and 4, is (Weisstein, 2002, p. 2970):

$$Volume = \left| \frac{Det\begin{pmatrix} x_1 & y_1 & z_1 & 1 \\ x_2 & y_2 & z_2 & 1 \\ x_3 & y_3 & z_3 & 1 \\ x_4 & y_4 & z_4 & 1 \end{pmatrix}}{3!} \right| =$$

$$x_1 * \left| \frac{Det\begin{pmatrix} y_2 & z_2 & 1 \\ y_3 & z_3 & 1 \\ y_4 & z_4 & 1 \end{pmatrix}}{3!} \right| - y_1 * \left| \frac{Det\begin{pmatrix} x_2 & z_2 & 1 \\ x_3 & z_3 & 1 \\ x_4 & z_4 & 1 \end{pmatrix}}{3!} \right| + z_1 * \left| \frac{Det\begin{pmatrix} x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \\ x_4 & y_4 & 1 \end{pmatrix}}{3!} \right| - 1 * \left| \frac{Det\begin{pmatrix} x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \\ x_4 & y_4 & z_4 \end{pmatrix}}{3!} \right| =$$

$$(x_1 * y_2 * z_3 * 1 - x_1 * 1 * y_2 * z_4 - x_1 * z_2 * y_3 * 1 + x_1 * z_3 * 1 * z_4 + x_1 z_2 * 1 * y_4 - x_1 * 1 * z_3 * y_4 -$$

$$- y_1 * x_2 * z_3 * 1 + y_1 * x_2 * 1 * z_4 + y_1 * z_2 * x_3 * 1 + y_1 * y_1 * z_2 * 1 x_4 + y_1 * 1 * x_3 * z_4 - y_1 * 1 * z_3 * x_4 +$$

$$+ z_1 * x_2 * y_3 * 1 - z_1 * x_2 * 1 * y_4 - z_1 * y_2 * x_3 * 1 + z_1 * y_2 * 1 * x_4 + z_1 * 1 * x_3 * y_4 - z_1 * 1 * y_3 * x_4 -$$

$$- 1 * x_2 * y_3 * z_4 + 1 * x_2 * z_3 * y_4 - 1 * y_2 * x_3 * z_4 - 1 * y_2 * z_3 * x_4 + 1 * z_2 * x_3 * y_4 - 1 * z_2 * y_3 * x_4)$$

$$* (1/6)$$

4.5 Using the ISBSG repository to set up performance targets

How can target results be set in the field of software engineering? For new organizations with no historical data, there is no easy way to set target results, and one possible solution is to use external performance data, such as those available in the ISBSG Repository. The process of setting targets using a benchmarking approach helps managers minimize risk and set performance targets objectively. The ISBSG Repository enables such benchmarking by giving managers a better understanding of what has happened in other organizations in order set or their own performance targets.

This section presents the MultiPERF prototype features that use the ISBSG Repository, as shown in Figure 4.41 and explained in:

- Section 4.5.1 – set performance targets;
- Section 4.5.2 – set organizational performance levels;
- Section 4.5.3 – select datasets that are comparable to those of the project at hand;
- Section 4.5.4 – use statistical analysis in order to better understand the selected datasets;
- Section 4.5.5 – build a model to set performance targets graphically.

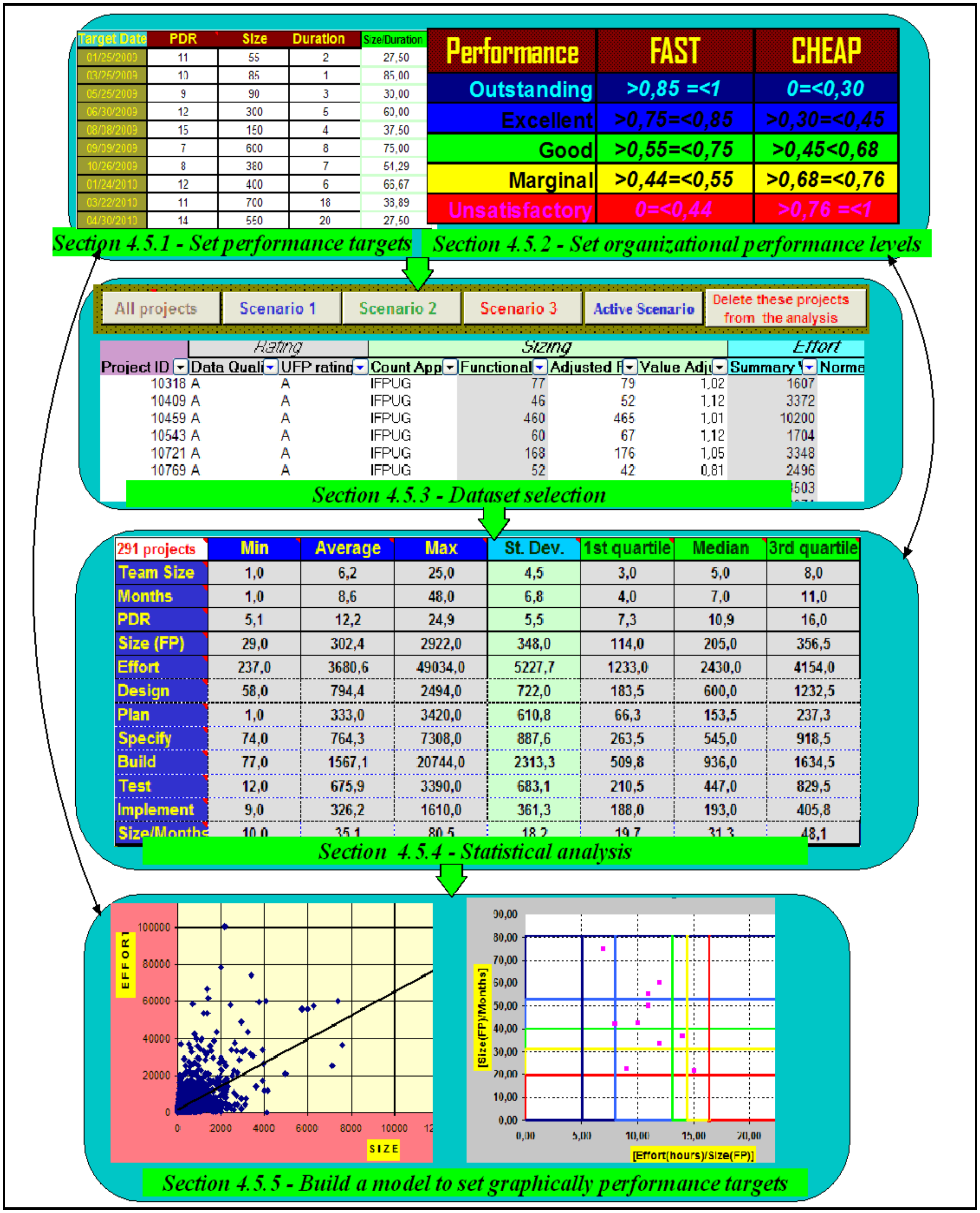


Figure 4.41 Using the ISBSG repository to set performance targets.

4.5.1 Setting performance targets

It is useful to be able to set performance targets based on information from different sources: other organizations in the same field (or in another field), customers, competitors, suppliers, standards, etc. As shown in Figure 4.45, the features in the MultiPERF prototype based on the ISBSG Repository can help the user set targets for effort, size, and duration, and ratios of these measures: productivity (Project Delivery Rate (PDR): Effort/Size) and Speed of Delivery (Size/Duration).

Setting performance target values for productivity and speed of delivery requires size, effort, and duration data from past projects:

- The manager must measure size in order to manage productivity. Determining the size of software is important, and is one of the first steps in the process of estimating a software development project. It is considered a primary input to most software development estimation models (Abran *et al.*, 2007; Abran *et al.*, 1998; Galorath and Evans, 2006; Lokan *et al.*, 2001). The ISBSG Repository contains data on the functional size of past projects.
- Effort corresponds to the amount of work expended to develop or maintain software, and an estimation of the effort to be expended is an important input to activity scheduling, investment decisions, budget determinations, and contract bidding. Effort is represented in total hours spent on a project.
- Duration models (Bourque *et al.*, 2007) are used to determine the time required to realize a project, which is shown in calendar-months in the ISBSG Repository. Duration covers all activities of the project and corresponds to the total elapsed time.

The example in Table 4.4 shows that performance targets can be broken down into iterations in MultiPERF for a single project:

- performance targets were set by the user for ten iterations in this example:
    - the target date for the first iteration is 02/15/2009 and the last target date is 04/30/2010;

- total projected functional size of the software to be delivered is 1000;
- the target total effort in hours is 19000;
- target total duration of the project is 15 months and the target PDR is 19 across all iterations;

- Calculated value: based on the PDR, Size, and Duration targets, the columns containing the Size/Duration and the Effort columns are calculated automatically.

Table 4.4 Target values per iteration for PDR, Size, Duration (Months), Size/Duration, and Effort

		Cumulative	Cumulative		Cumulative
Target Date	PDR	Size	Duration	Size/Duration	Effort
02/15/2009	19	10	1	10,00	190
03/25/2009	19	55	2	27,50	1045
05/25/2009	19	75	4	18,75	1425
06/30/2009	19	125	5	25,00	2375
08/08/2009	19	175	7	25,00	3325
09/09/2009	19	225	8	28,13	4275
10/26/2009	19	300	9	33,33	5700
01/24/2010	19	500	12	41,67	9500
03/22/2010	19	700	14	50,00	13300
04/30/2010	19	1000	15	66,67	19000

The example in Table 4.5 shows the performance targets set by the user for ten projects:

- the target date for the first project is 02/15/2009 and the target date for the last one is 04/24/2011;
- the functional size of the project software to be delivered is between 525 and 15300;
- the total effort in hours is between 525 and 15300;
- minimum duration for the project is 1 month and the maximum is 35 months;

- Calculated value: based on the PDR, Size, and Duration targets, the columns containing the Size/Duration and the Effort columns are calculated automatically.

Table 4.5 Target values per project for PDR, Size, Duration (Months), Size/Duration, and Effort

Target Date	PDR	Size	Duration	Size/Duration	Effort
02/15/2009	12	75	1	75,00	900
03/25/2009	25	237	2	118,50	5925
05/25/2009	7	75	4	18,75	525
06/30/2009	17	125	3	41,67	2125
08/08/2009	8	175	7	25,00	1400
09/09/2009	20	225	8	28,13	4500
10/26/2009	19	300	9	33,33	5700
01/24/2010	16	500	12	41,67	8000
03/22/2012	18	850	35	24,29	15300
04/24/2011	9	700	26	26,92	6300

4.5.2 Setting the organizational performance level

Five levels of performance can be defined by the user, as shown in Figure 4.42. These levels are represented on the performance-benchmarking graph shown in Figure 4.47:

- Outstanding – above the blue line. The best performance of all is the minimum value on the x axis and the maximum value on the y axis – represented by a dark blue line;
- Excellent – above the target performance level, but still not outstanding – between the green line and the blue line;
- Good – as expected – between the yellow line and the green line;
- Marginal – less than the target performance, but within acceptable limits – between the red line and the yellow line;
- Worst – unacceptable – between zero and the red line.

All performance values included in the Fast and Cheap columns are normalized on a scale from zero to one. In the Cheap column is the ratio of effort in hours to size, as shown in Figure 4.42. In the Fast column is the ratio of size to duration in months, as shown in Figure 4.42. As shown in Figure 4.47, the scale from zero to one is automatically transformed as a function of dataset values:

- Cheap – The project with the lowest ratio (of effort in hours to size) in the dataset will have a value of zero (cheapest project or project with the highest productivity ratio), and the maximum value will be equal to one (most expensive project or project with the lowest productivity ratio).
- Fast – The maximum value of the ratio of size to duration in months in the dataset will have value of one (fastest project or project with the highest speed of delivery ratio), and the minimum value will be equal to zero (slowest project or project with the lowest speed of delivery ratio).

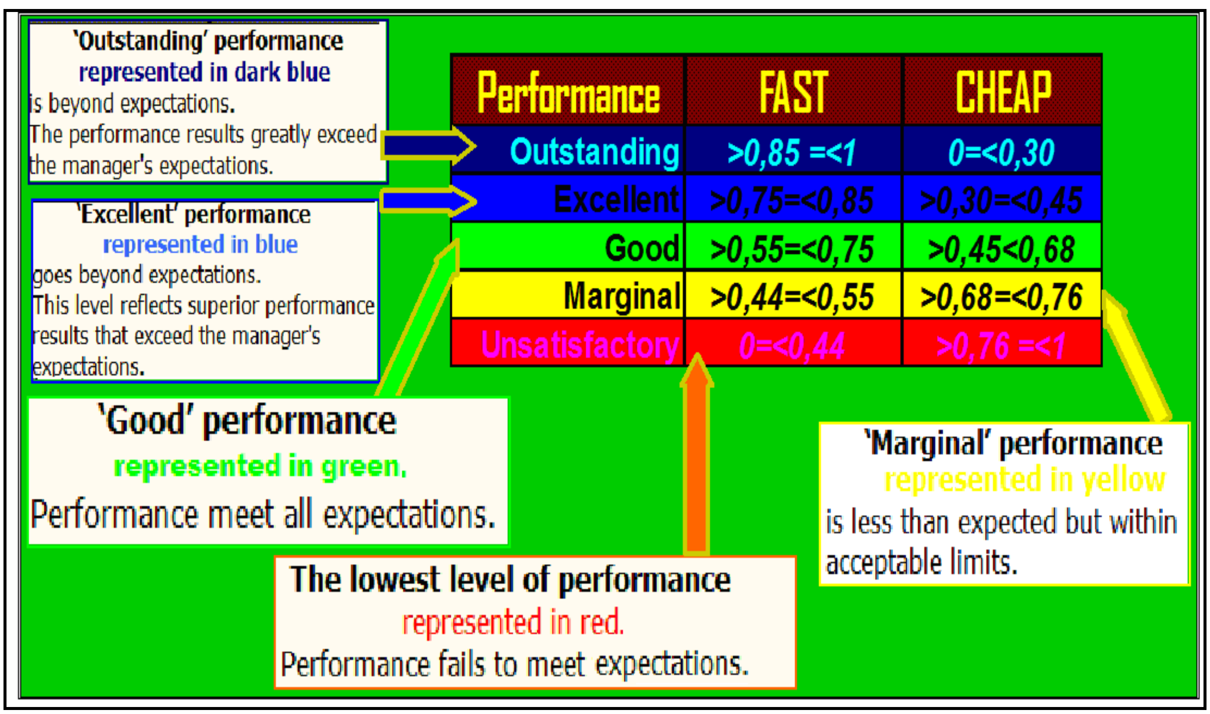


Figure 4.42 Target performance levels for benchmarking purposes.

4.5.3 Selecting the dataset

A manager can choose the most appropriate dataset for a particular condition or scenario in their organization, as shown in Figure 4.43:

- There are more than 100 variables in the Repository, and a manager can choose the most appropriate projects and variables for his situation. Every organization is different, and every project is different, and so there will always be different variables that must be taken into account.
- A short text is available explaining every variable. In addition, every variable has a filter feature with a drop-down list containing all the values available for that variable.
- There is the possibility of storing three selected datasets for later use (Scenario 1, Scenario 2, and Scenario 3), as shown for Scenario 1 in Figure 4.43. Also, an Active Scenario will show all the active filters for the active dataset.
- Statistical tables and graphs are automatically updated.
- Other projects from the user's own organization or from other organizations can be added to the Repository.

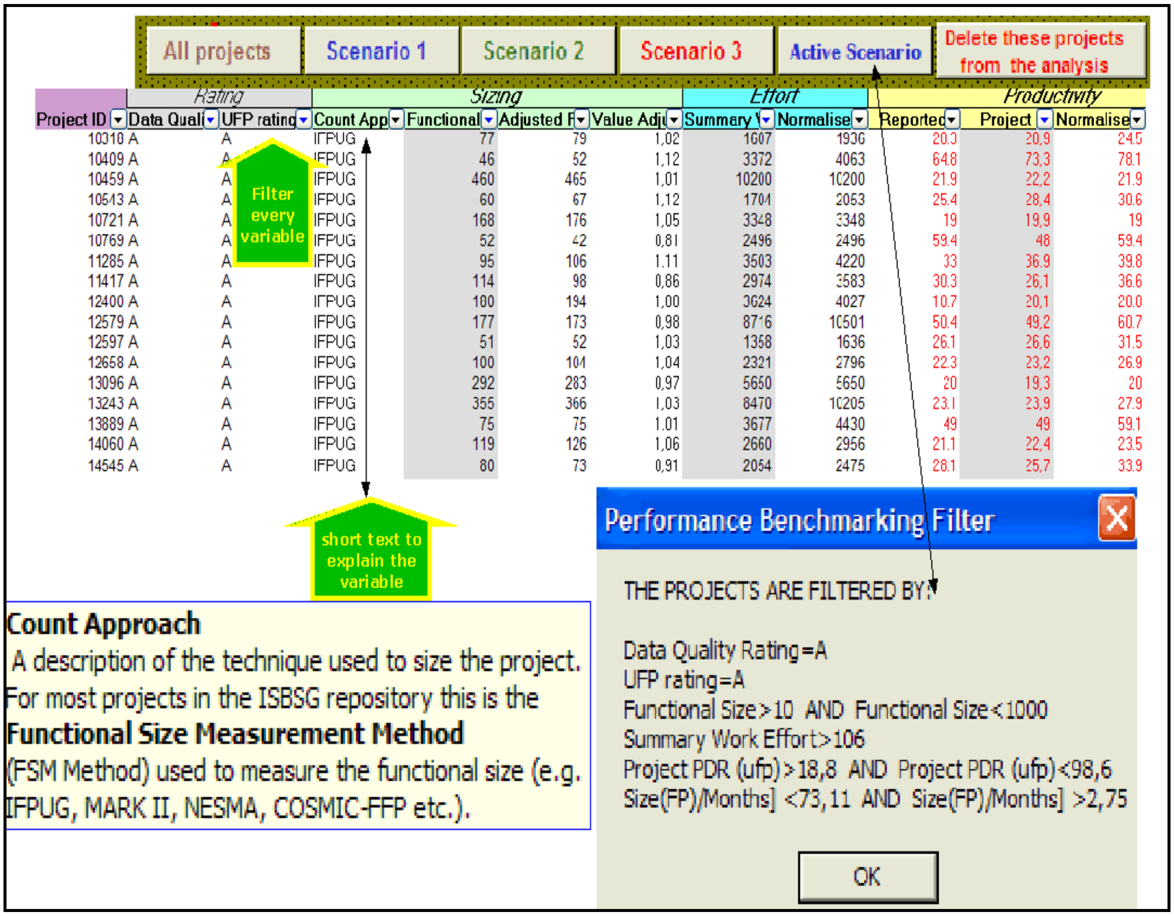


Figure 4.43 Selecting a dataset from the ISBSG repository.

4.5.4 Statistical analysis

Table 4.6, an example generated using the prototype, shows that different results of predefined descriptive statistical functions can be produced for the dataset selected by the user.

The ISBSG variables used to describe the dataset statistically as is shown in Table 4.6 are:

- **Teams size -** the maximum number of people that worked at any time on the project.
- **Months (Duration)** – the number of months it took to complete a project;

- **PDR (P**roject **D**elivery **R**ate) – calculated as **Effort / Size** and used as a measure of productivity: high values mean low productivity;
- **Size** – reported in different units, depending on the functional sizing method adopted: IFPUG, NESMA, MARK II, or COSMIC-FFP, as shown in the fourth column of Figure 4.43;
- **Effort** – referred to as Summary Work Effort, indicates the total effort in hours recorded against the project;
- **Design** – the breakdown of the work effort reported for the Design phase;
- **Plan** – the breakdown of the work effort reported for the Plan phase;
- **Specify** – the breakdown of the work effort reported for the Specification phase;
- **Build** – the breakdown of the work effort reported for the Development phase;
- **Test** – the breakdown of the work effort reported for the Test phase;
- **Implement** – the breakdown of the work effort reported for the Implementation phase;
- **Size/Months** – the ratio of size to duration**.**

The definitions of statistical terms used in the Table 4.6 are the following:

- **Projects** – the number of projects in the selected dataset; as shown in the top-left part of the table, there are 291 projects;
- **Min** – the minimum value found in the selected dataset;
- **Average** – all the values added together divided by how many values have been added;
- **Max** – the maximum value found in the selected dataset;
- **Std Dev** – the standard deviation, which shows the variation among the projects in the dataset;
- **1st quartile** – the 25th percentile: "the value one-quarter of the way through the ordered set of data" (Morris, 2003, p. 119);
- **Median** – the middle value, representing the value that delimits half the projects in the dataset: half the values are below and half the values are above this median;
- **3rd quartile** – the 75th percentile: "the value three-quarters of the way through the ordered set of data" (Morris, 2003, p. 119).

Table 4.6 Descriptive statistics generated

291 projects	Min	Average	Max	St. Dev.	1st quartile	Median	3rd quartile
Team Size	1,0	6,2	25,0	4,5	3,0	5,0	8,0
Months	1,0	8,6	48,0	6,8	4,0	7,0	11,0
PDR	5,1	12,2	24,9	5,5	7,3	10,9	16,0
Size (FP)	29,0	302,4	2922,0	348,0	114,0	205,0	356,5
Effort	237,0	3680,6	49034,0	5227,7	1233,0	2430,0	4154,0
Design	58,0	794,4	2494,0	722,0	183,5	600,0	1232,5
Plan	1,0	333,0	3420,0	610,8	66,3	153,5	237,3
Specify	74,0	764,3	7308,0	887,6	263,5	545,0	918,5
Build	77,0	1567,1	20744,0	2313,3	509,8	936,0	1634,5
Test	12,0	675,9	3390,0	683,1	210,5	447,0	829,5
Implement	9,0	326,2	1610,0	361,3	188,0	193,0	405,8
Size/Months	10,0	35,1	80,5	18,2	19,7	31,3	48,1

4.5.5 Building a model to set performance targets graphically

The first step in building a model to help the user set targets is to identify the requirements of the software to be built, in order to estimate the functional size accurately, as shown in Figure 4.45. Then, the variables that can affect effort and duration must be identified (business sector, application language, hardware platform, or use of case tools, for example). Finally, once the variables have been identified, various graphical models can be analyzed, as well as the results of the predefined statistical functions in Table 4.6, as shown in Figure 4.44, Figure 4.45, Figure 4.46, and Figure 4.50.

Size is important in estimation, and constitutes a base input to most software projects. As shown in Figure 4.45 and Figure 4.46, the graphs available using size are: Effort vs. Size, Duration vs. Size, and Productivity vs. Size.

Effort is represented in total hours spent on a project, and estimating the target value of this measure well is critically important. Using the prototype, it is possible to track the amount of effort expended in the design, plan, specification, development, test, and implementation phases, as shown in Table 4.6 (Déry and Abran, 2005). The technique chosen in the prototype to determine target Effort is to build estimation models using regression equations, as shown in Figure 4.44, Figure 4.45, and Figure 4.46. The database can be divided into two or more smaller datasets to be analyzed separately, if there are enough projects after division. As shown in Figure 4.45, project Size is a major input for project Effort, and the manager in this case has set the final size for the target at 1000 as is shown in Table 4.4. For example, the first dataset could be the result of using the filter for sizes from 10 to 1000, as shown in Figure 4.46, where there are 2105 projects. As shown in Figure 4.45 and Figure 4.46, the graphs available that use Effort are: Effort vs. Size, and Effort vs. Duration.

As project Duration determines the time required to realize a project, and covers all activities of the project, the Effort will represent a major input. Also, Duration is affected directly by Productivity, as shown in Figure 4.45, where Productivity = Effort/Size. As shown in Figure Figure 4.45, and Figure 4.46, the graphs available using Duration are: Effort vs. Duration, and Duration vs. Size.

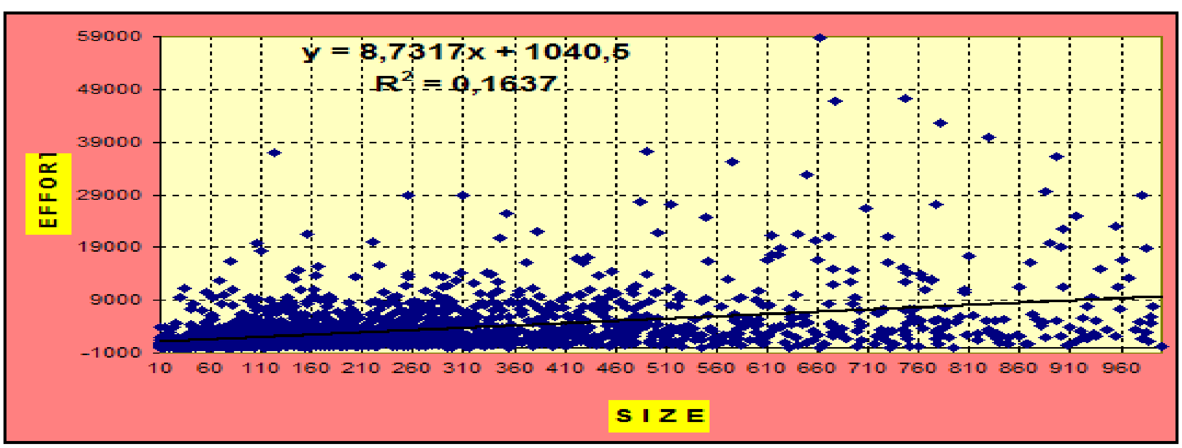


Figure 4.44 Productivity model – Regression analysis (ISBSG).

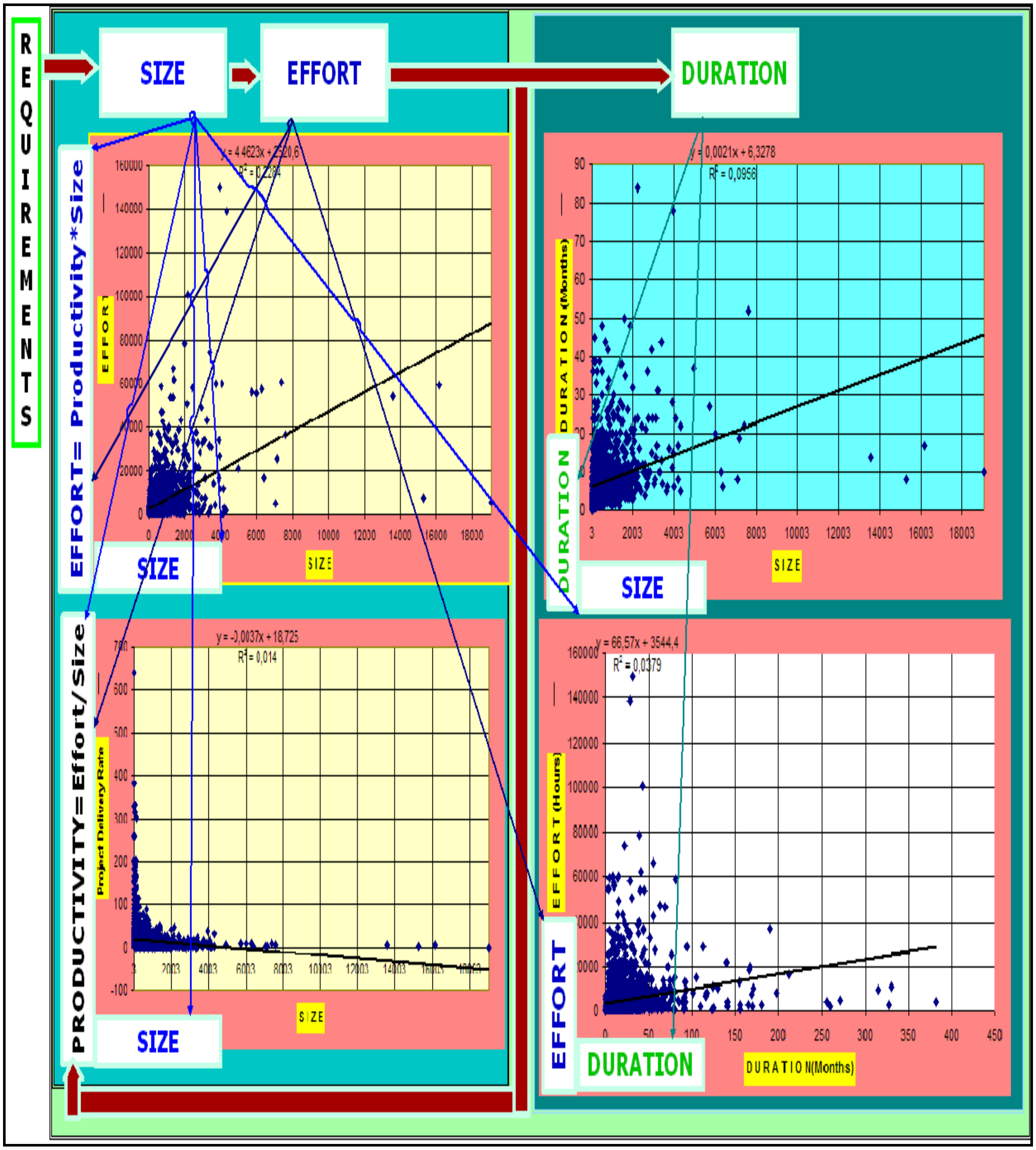


Figure 4.45 Estimating size, effort, productivity, and duration.

2105 projects	Min	Average	Max	St. Dev.	1st quartile	Median	3rd quartile
Team Size	0,5	6,3	309,0	12,1	2,0	4,0	7,0
Months	0,0	6,7	48,0	5,5	3,0	5,0	8,5
PDR	0,1	17,4	382,2	26,8	4,5	9,7	20,2
Size (FP)	10,0	254,1	999,0	220,5	89,0	178,0	354,0
Effort	0,0	3259,5	58700,0	4759,2	701,0	1704,0	3890,0
Design	17,0	855,0	5807,0	1172,0	183,5	600,0	1232,5
Plan	1,0	300,1	10800,0	828,9	21,0	94,0	304,0
Specify	1,0	1161,5	28153,0	2232,1	157,3	520,0	1245,0
Build	1,0	1353,3	32340,0	2466,9	156,0	624,0	1545,0
Test	1,0	681,2	14196,0	1194,8	61,8	247,5	747,5
Implement	1,0	307,7	16136,0	1006,1	44,0	92,0	232,5
Size/Months	0,0	46,2	897,3	57,9	11,6	29,7	60,5

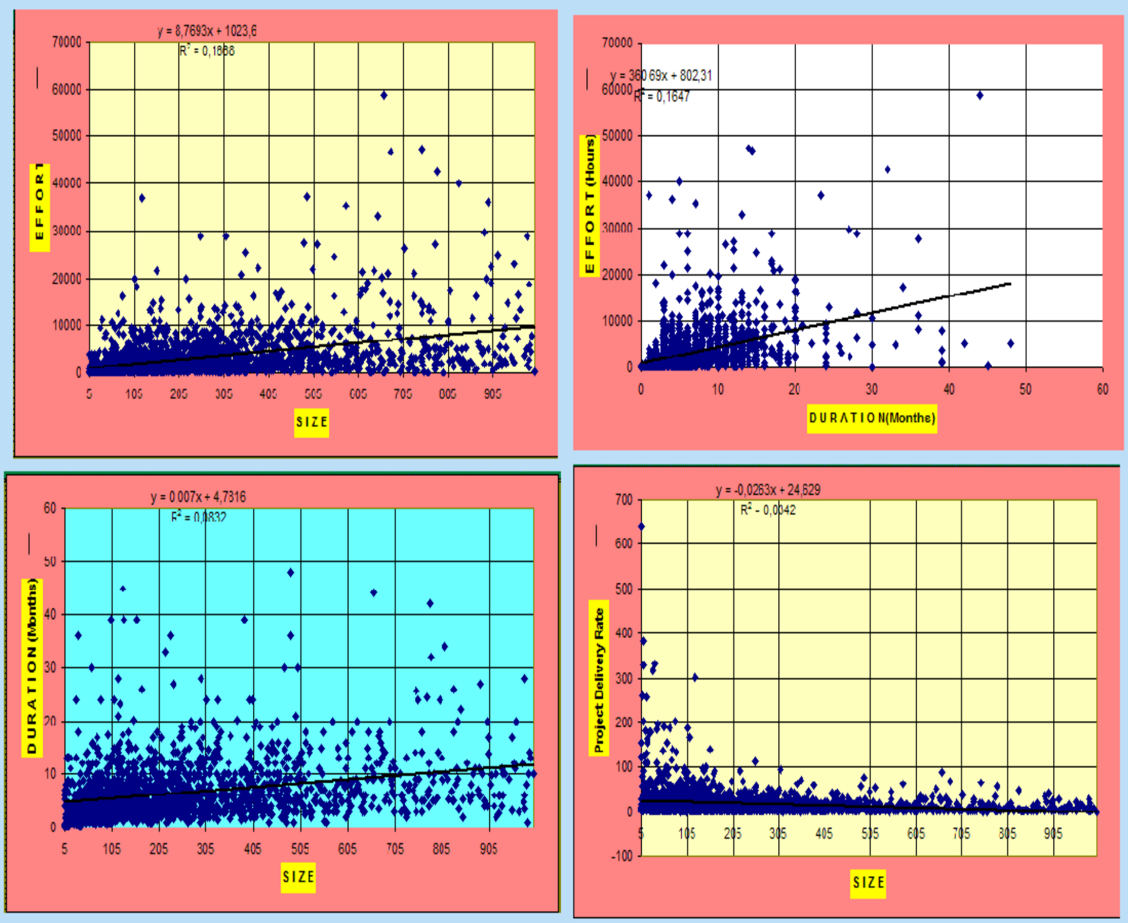


Figure 4.46 Statistical analysis and graphical models.

One way to determine whether targets are reasonable is by benchmarking them against a representative dataset selected from the ISBSG Repository, as shown in Figure 4.50:

- The x axis represents the ratio between effort and size:
    - The cheapest projects have the lowest value ratios;
    - The most expensive projects have the highest value ratios;
    - As shown in Figure 4.42, five levels of performance are defined by the user and are represented on the graph using the five vertical color lines.
- The y axis represents the ratio between size and duration:
    - The fastest projects have higher value ratios;
    - The slowest projects have lower value ratios;
    - As shown in Figure 4.42, five levels of performance are defined by the user and represented on the graph using the five horizontal color lines.
- The colors have the same meaning on both the x and y axes: dark blue (outstanding), blue (excellent), green (good), yellow (marginal), and red (unacceptable).
- This graph was inspired by the TreeMaps concept (HCIL, 2010), which consists of a series of 25 nested rectangles to represent the targets, as shown in Figure 4.48.
- How the targets defined by the user are represented on the graph is shown in Figure 4.49. The user can then determine whether or not this makes sense in their context. For example, if the target is classified as being among the outstanding projects in the selected dataset, then the user can analyze whether or not they can develop software at such a high speed of delivery and with such a high productivity rate. Does the organization have the know-how, experience, and maturity to be able to do this? Conversely, if the target is categorized among the unacceptable projects, then perhaps the target should be adjusted to take into account the particular features of the software to be developed, the development organization, and the similarity of the selected dataset to the software development project being estimated.
- The fastest and cheapest projects are situated in the upper-left corner of the graph, as shown in Figure 4.48, and the slowest and most expensive projects are situated on the opposite side, in the lower-right corner.

- The average projects are situated in a rectangle formed by the green and yellow lines, which is, not necessarily, in the center of the graph. The size of this rectangle depends on the dataset analyzed and the performance values in the Fast and Cheap columns, as shown in Figure 4.42.
- The marginal projects are situated in the rectangle between the red and yellow lines, which is, as expected, near the lower-right corner. In Figure 4.49, one of the targets is in this rectangle.
- Finally, the 25 nested rectangles delimit 25 distinct performance areas.

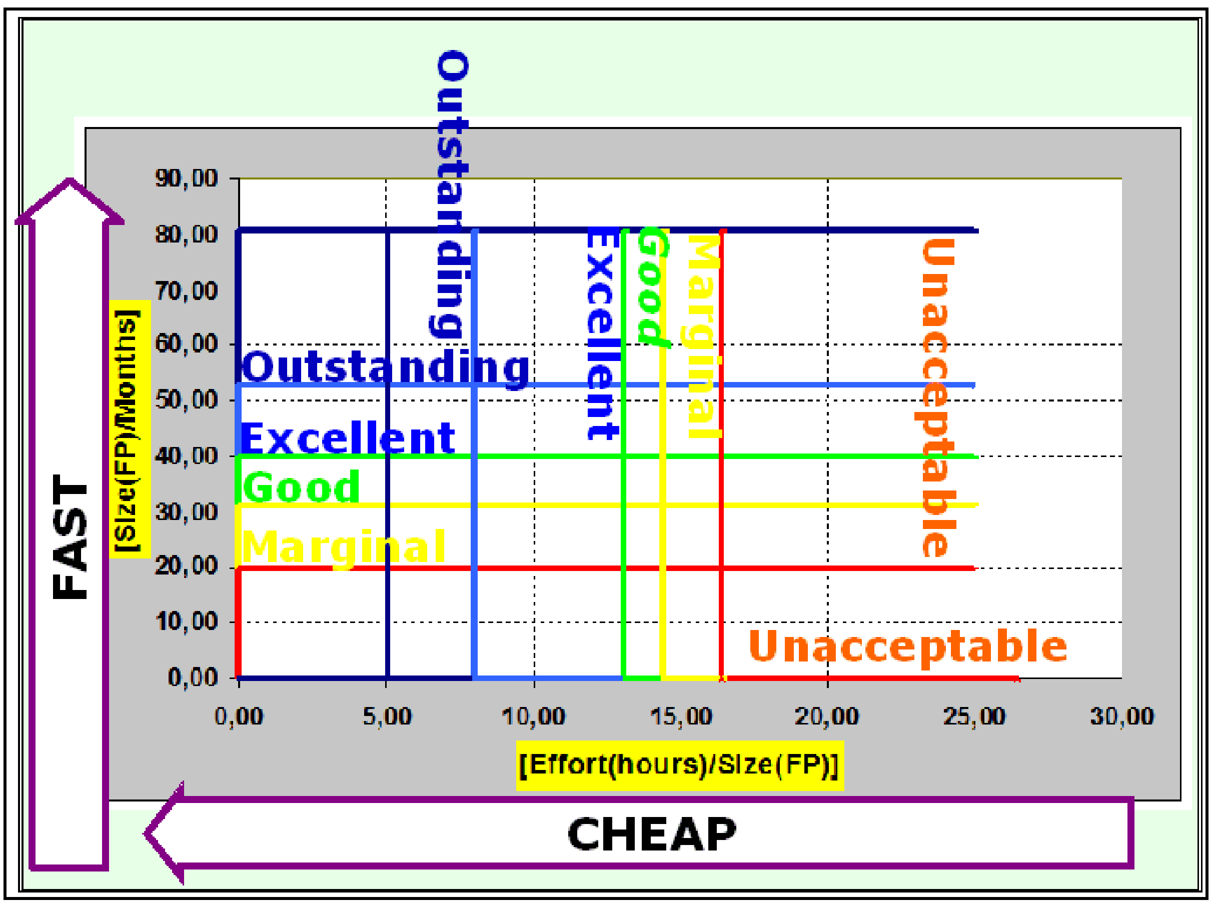


Figure 4.47 Benchmarking – Performance level representation.

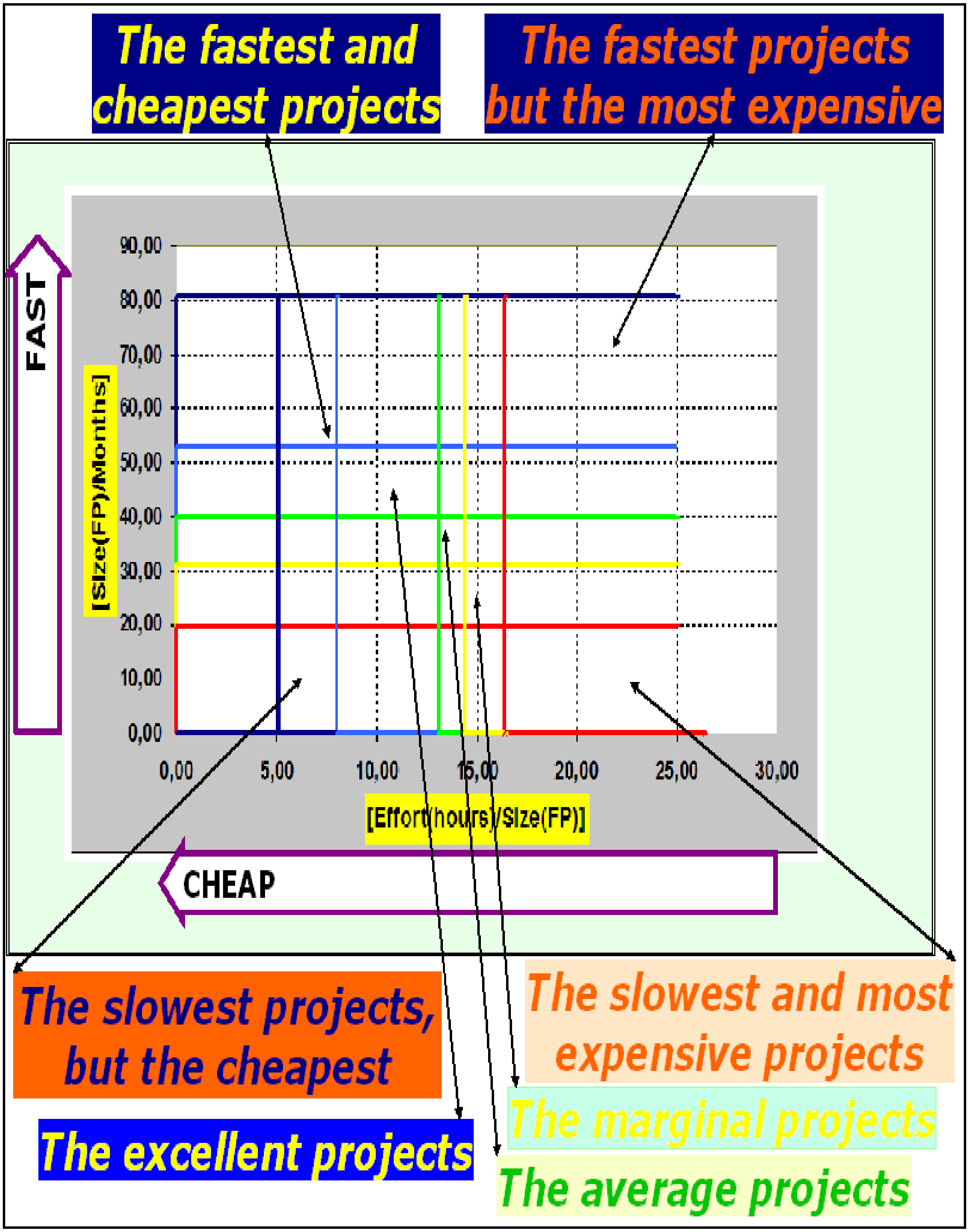
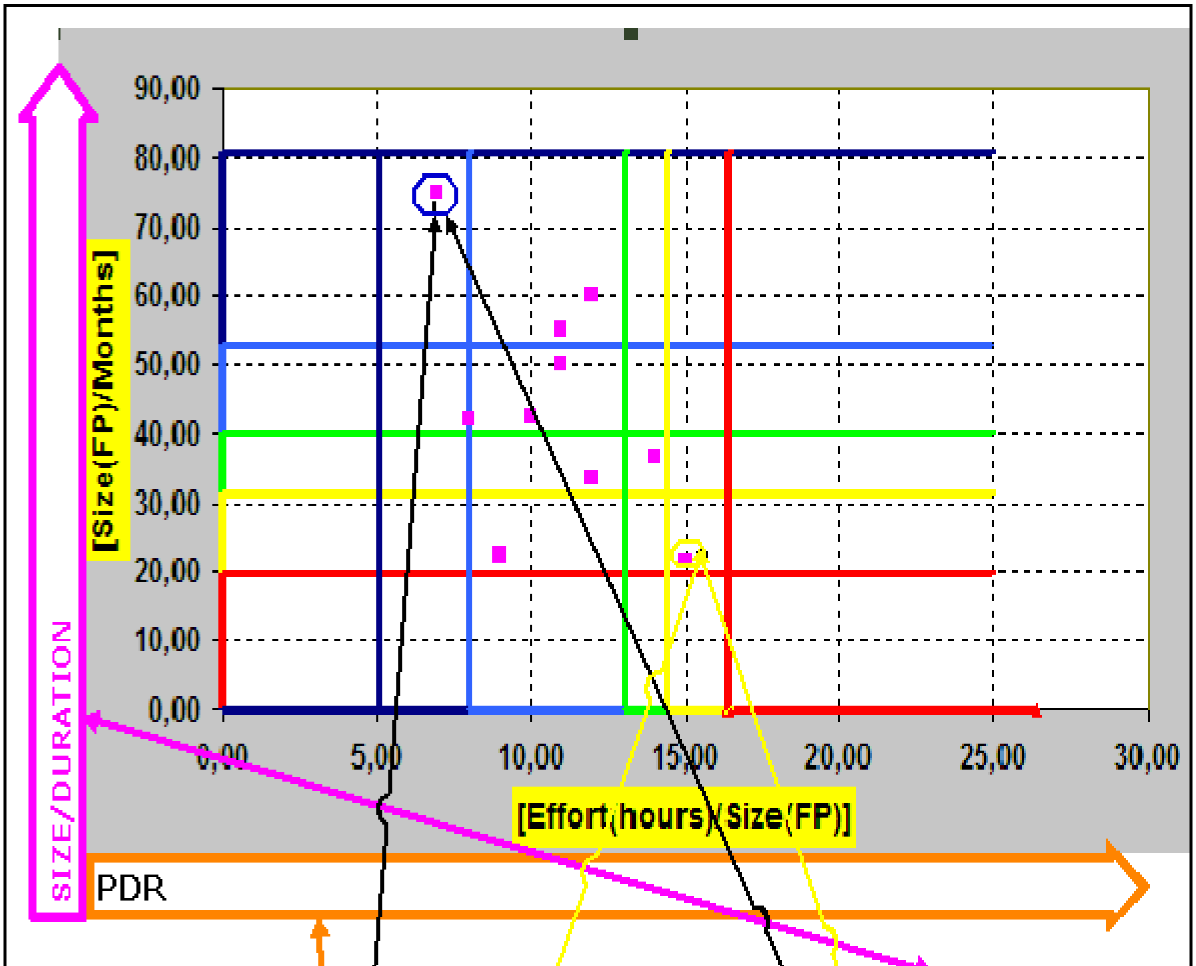


Figure 4.48 Performance benchmarking using the TreeMap concept.



Target Date	PDR	Size	Duration	Size/Duration
01/25/2009	11	55	1	55,00
03/25/2009	10	85	2	42,50
05/25/2009	9	90	4	22,50
06/30/2009	12	300	5	60,00
08/08/2009	15	150	7	21,43
09/09/2009	7	600	8	75,00
10/26/2009	8	380	9	42,22
01/24/2010	12	400	12	33,33
03/22/2010	11	700	14	50,00
04/30/2010	14	550	15	36,67

Figure 4.49 Target objectives, as defined in Table 4.5.

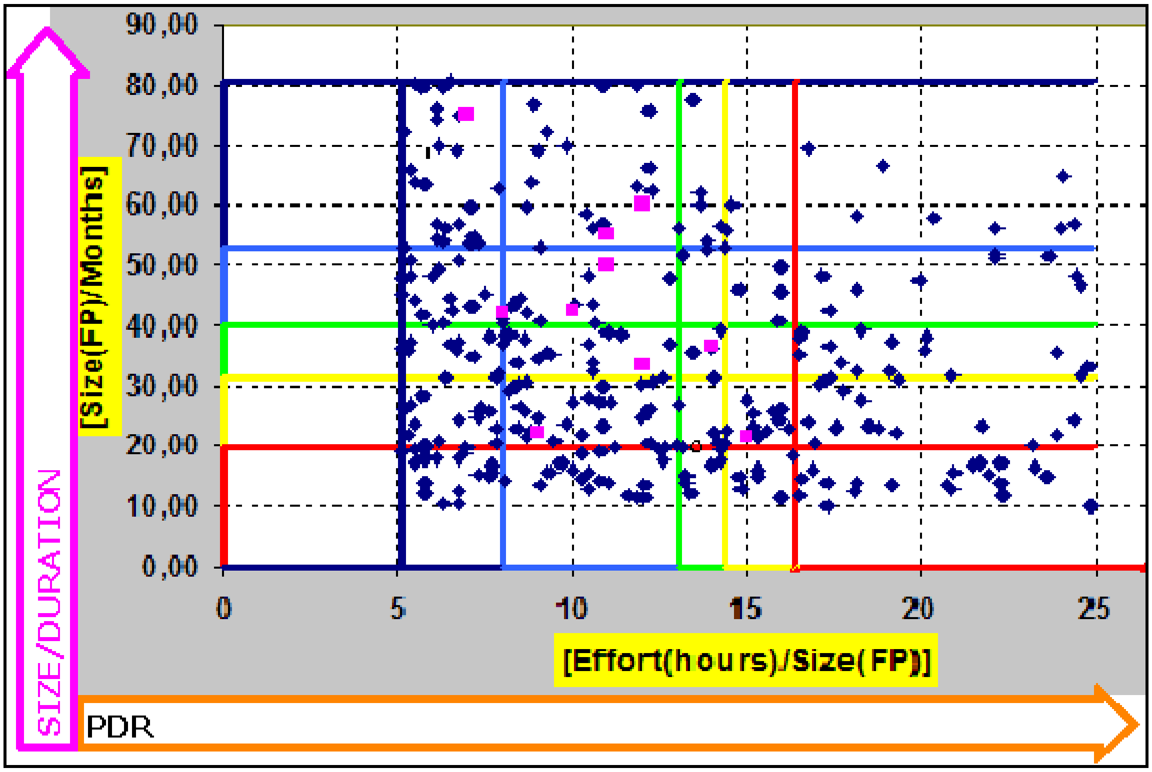


Figure 4.50 Target objectives, as defined in Table 4.5 vs. an ISBSG dataset.

4.6 Conclusion

A mapping between the conceptual framework and the prototype is shown in Figure 4.51:

- Section 4.2 discussed how the Plan-Design step, as presented in section 3.1.1, is implemented in the prototype;
- Section 4.3 described how the Internal and External database, as presented in section 3.1.2, is implemented using the ISBSG Repository and an internal database;
- Section 4.4 presented the basic features and the more advanced analytical tools, and described how the performance data are used, as presented in section 3.1.3;
- Section 4.5 described how benchmarking is used to compare organizational performance results with the ISBSG Repository, as presented in section 3.1.3 and section 3.1.4;
- A mapping between the conceptual framework and the prototype is shown in Figure 4.51.

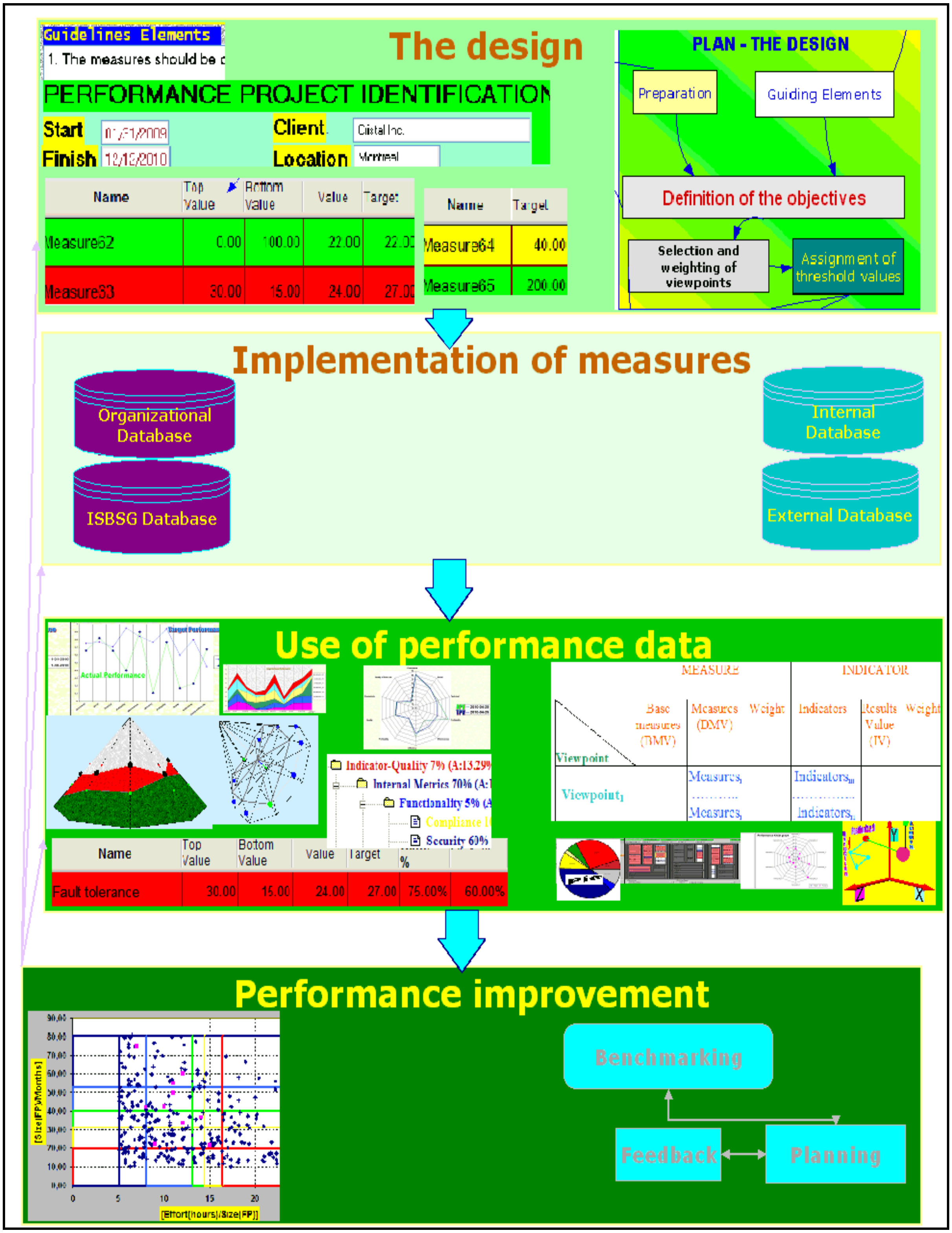


Figure 4.51 Prototype – Summary.

CONCLUSION

In this section, and as shown in Annex I, Figure-A I-1 the thesis is summarized, the limitations of the research are described, the answers to the research questions are shown diagrammatically (as in Annex II, Figure-A II-1), the contributions are presented, and suggestions for future research are made.

Summary

Managing performance is an important, and difficult, topic, and tools are needed to help organizations manage their performance. This thesis argues that visualizing, understanding, and improving organizational performance, which is multidimensional by definition, is an important problem, and a challenging one.
Performance Management Frameworks have become important to organizations that need to plan, monitor, control, and improve their decisions, their productivity, their efficiency, their effectiveness, the quality of their products, and their ability to deliver them on time. Organizations must measure and manage performance.

Several PMFs have been developed in different fields to address the management of organizational performance. However, they have a common role, which is to assure managers that organizational objectives are being achieved and that performance is improving.

One of the most important advantages of adopting and using a PMF is to be able to clearly differentiate organizational success from failure. Use of a PMF can show an organization how it is performing and indicate whether or not an organization is going in the right direction to achieve its objectives. Managing organizational performance is an important task that needs to be better understood, and this can be achieved by analyzing indicators using advanced techniques. Specifically, in the area of software engineering, a significant proportion of software development projects fail (DeMarco, 1982; Ewusi-Mensah, 2003;

Glass, 2005; Taylor, 2000; The Standish Group, 2009), and so measuring and managing performance is necessary in order that problems can be corrected as soon as possible in the early development stages, rather than when it is too late. Managing performance in software engineering is a complex task, as performance management varies widely from one organization to another and is, in many ways, unique to each individual software engineering organization. Therefore, performance management tools are needed that take into account the specifics of software development and maintenance as well as the specifics of each organization.

There are many frameworks available in the area of performance management, and their purpose is not only to simply measure performance, but also, more comprehensively, to manage it. Performance management models and frameworks vary widely in terms of the viewpoints they consider, and the indicators or measurements within each viewpoint. The frameworks presented in chapter 2 show clearly that performance is not one-dimensional, and it is essential that it be viewed from multiple and interlinked viewpoints.

PMFs must take into consideration the values of the actual performance and the target values for each measure, indicator, and viewpoint. Organizations vary considerably in the wide variety of dimensions that influence their performance, and every organization has their own viewpoints (economic, social, technical, customer satisfaction, risk, etc.) that they want to manage, and which must be represented in a consolidated manner in order to manage organizational performance overall.

The goal of this thesis is to develop a software prototype for multidimensional performance management in software engineering. The main objective of the research is to develop a prototype for multidimensional performance management in software engineering which can represent, quantitatively and in consolidated manner, the many possible performance viewpoints, while at the same time tracking the values of the individual performance dimensions.

This thesis proposes a conceptual framework and a prototype for managing performance in a multidimensional manner in the area of software engineering management. In the first phase, the first two chapters reviewed and analyzed the performance models and performance management frameworks that can be found in the literature, presented the International Software Benchmarking Standard Group (ISBSG) Repository, and analyzed the QEST prototype. In addition, the important terms of this research were identified and defined.

In the second phase, in chapter 3, a proposal for a conceptual framework for managing organizational performance in the context of software engineering was developed. The proposed framework is divided into four steps: the plan-design step, the implementation of measures step, the use of the framework step, and the performance improvement step. The framework is based on an integrated view of organizational performance, and provides a structured way of managing performance in a methodical manner which is adaptable to a specific organization.

In the third phase, in chapter 4, the MultiPERF prototype, developed to manage multidimensional performance in software engineering, is presented and discussed. The prototype can be used for multidimensional performance management in software engineering, either at the organizational level or at the project level. It adopts a multidimensional geometrical approach using advanced techniques inspired by the QEST model. The prototype includes visual analytical tools to manage, interpret, and understand the results in a consolidated manner, at the same time keeping track of the values of the individual dimensions of performance. A benchmarking approach using the ISBSG Repository is also included in the prototype to enable the user to set or determine appropriate performance targets (presented in section 4.5). Essentially, the prototype implements the components of the framework outlined in chapter 3, as shown in Figure 4.51.

Finally, the fourth and final phase is presented in this chapter, where the research is summarized, the limitations and contributions are described, and suggestions for future research are put forward.

Limitations

A summary of the limitations of this thesis regarding the literature review, the viewpoints, the indicators and measures, the external database, the conceptual framework, and the prototype as is presents and, are shown diagrammatically (as in Annex I, Figure-B I-2).

Literature review

The thesis presents and analyzes the multidimensional performance management frameworks that are currently available to management generally, and to software engineering management specifically. There are many PMFs presented in the literature, and it was necessary for reasons of scope to limit the analysis to the selected subset of multidimensional performance management models or frameworks.

The definitions of key concepts that were presented were selected from the many definitions available. In addition, only definitions for the terms included in the thesis title were presented.

Viewpoints, Indicators, and Measures

There is no limit to the number of measures and indicators for the prototype, nor are there any limits on the formula used to calculate and represent viewpoints. However, the prototype was tested by the author using 10 viewpoints.

External Databases

This thesis incorporates only one repository of software project data. Other repositories can include other measures and variables not included in our prototype.

Performance

It is important to understand that this prototype and its associated conceptual framework is limited to organizational performance, and does not cover any other type of performance, such as individual performance or the performance of software in terms of memory usage, transaction time, and so forth. Also, the prototype was developed to manage performance in only one organization.

Framework

The prototype and framework presented in this thesis have never actually been used in industry, so this is probably the most important limitation of this research. Industrial usage and adoption are obviously important; however, it was not included as a research objective for reasons of scope. It is proposed, of course, as a suggestion for future research.

Answers to Research Questions

Specific sections of this thesis are directly linked to the research questions that were presented in the Introduction (as is shown in Annex II, Figure-A II-1). The boxes (yellow background color) at the top of Figure-A II-1 contain the research question numbers and the second-level boxes (yellow background color as well) contain the actual research questions. The bottom-level boxes (white background color) indicate the sections in this thesis where answers to these questions can be found.

In Phase I of this thesis, the models and frameworks found in the literature were reviewed, and the QEST model and the International Software Benchmarking Standard Group (ISBSG) Repository were analyzed. Chapters 1 and 2 therefore provide a detailed answer to Parts A and B of the first research question respectively. The focus of section 2.5 is on answering the second research question specifically.

The focus of Phases Two and Three is on providing an answer to the third research question, which is related to the development of a prototype for multidimensional performance management in software engineering that can represent, graphically and in a consolidated manner, the many possible performance viewpoints, while at the same time keeping track of the values of the individual dimensions. In this regard, a conceptual framework was proposed in chapter 3, and a prototype called 'MultiPERF' was developed and is discussed in chapter 4 to manage multidimensional performance in software engineering based on this framework. The framework and prototype integrate the concepts from various sources that were analyzed and discussed in Phase I.

Contributions

The main research contributions of this dissertation are a conceptual framework, and the development of a prototype supporting this framework, to manage performance in the field of software engineering.

A proposal for a conceptual framework for managing organizational performance in the field of software engineering was developed, and is presented in chapter 3. The framework, based on an integrated view of organizational performance, provides a structured way of managing performance in a methodical manner and is adaptable to a specific organization. The framework is made up of four steps:

- Plan-Design step – an important phase in order to decide what elements of performance to measure and how to measure it;
- Implementation of measures step – a difficult phase for an organization, because it "can be seen as 'changing the rules of the game' or 'redistributing power in the organization'" (Bourne *et al.*, 2000);
- Use of the framework step – visualization, analysis, and interpretation of the results: using appropriate graphical tools to have the right information at the right moment;
- Performance improvement step – provide feedback to enable performance improvement and facilitate benchmarking.

A prototype is presented in chapter 4 that supports or implements the conceptual framework discussed in chapter 3. The prototype integrates concepts from various sources, and adopts:

- a consolidated view of organizational performance using a variety of graphical and statistical analysis tools;

- a comprehensive view of organizational performance using a multidimensional geometrical approach inspired by the QEST model;
- a set of indicators and measures from ISO-9126, which is a standard used to describe the quality of software;
- the Sink and Tuttle viewpoints;
- a drill-down and drill-up feature to analyze and visualize different levels of granularity of performance;
- the ISBSG Repository to help define performance constraints;

Future Research Suggestions

The research presented in this thesis on a conceptual framework to manage performance and a prototype in support of this conceptual framework raises issues which can be pursued in future research.

The following research directions are proposed: use of the conceptual framework and the prototype in industry, adaptation of the prototype to other fields, improvement of the conceptual framework, and incorporation of other geometrical approaches and mathematical formulae.
The proposed framework and prototype in the industry to manage multidimensional performance requires verification of both in actual settings.

The prototype was developed for managing performance in the field of software engineering. However, it could potentially be used in another field. The application of the framework and the prototype in other fields may, however, require modification and adaptation of the framework and the prototype to:

- integrate indicators and measures from standards used in other fields;
- adapt to other sources of external public or private performance data than the ISBSG Repository;
- add other geometrical approaches;

- include statistical analyses and graphs used in other fields.

Of course, benchmarking models for analysis and to set performance targets in other fields may require other kinds of analysis.

The framework and prototype developed in this work concentrate on managing performance in only one organization. Certainly, modifications would be required in order to use the framework and the prototype for managing performance in more than one organization, or in multiple divisions of the same organization at a time, and for performing inter-organizational analysis.

ANNEX I

RESEARCH PHASES AND SUMMARY OF THE LIMITATIONS

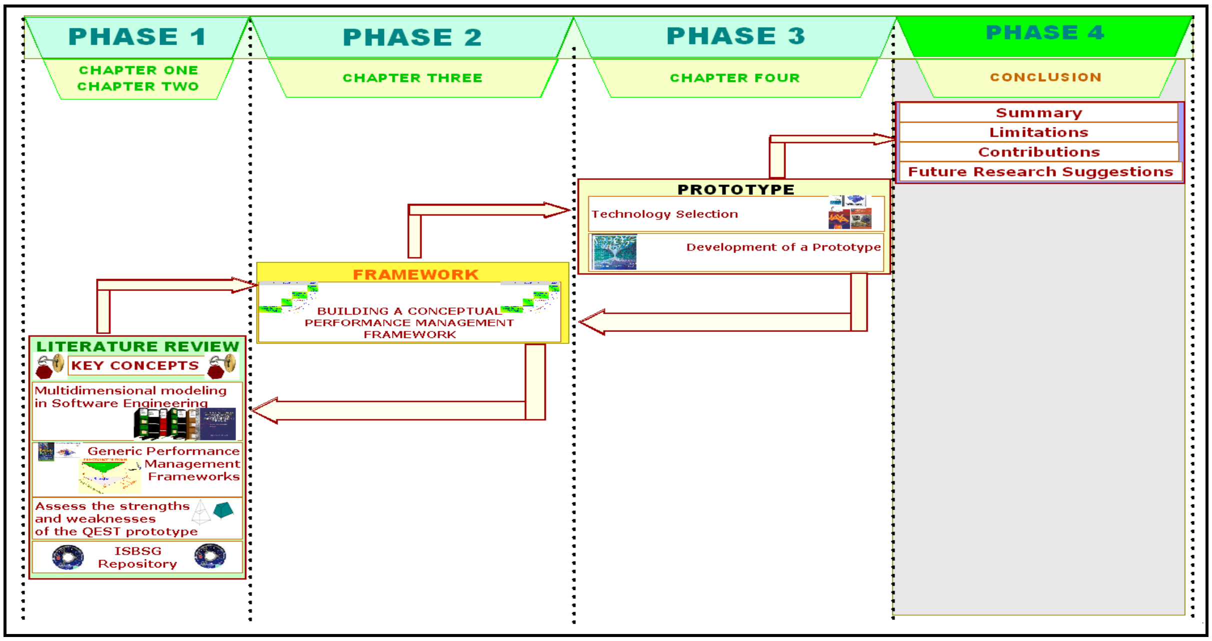


Figure-A I-1 Development of a Prototype – Conclusion.

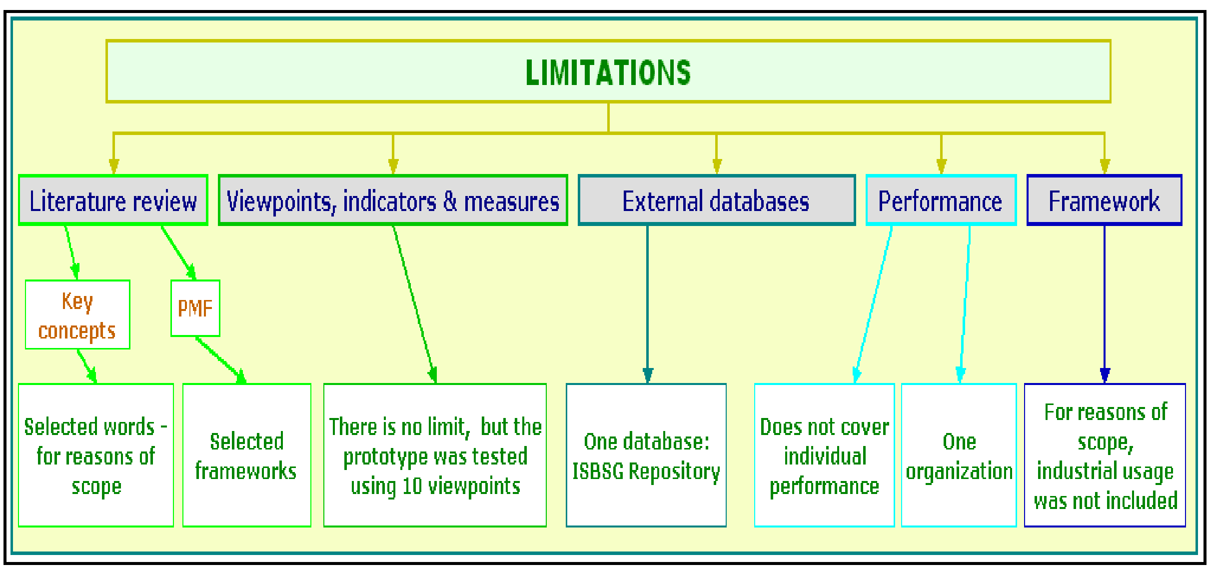


Figure-B I-2 Summary of the limitations of this thesis.

ANNEX II

ANSWERS TO RESEARCH QUESTIONS

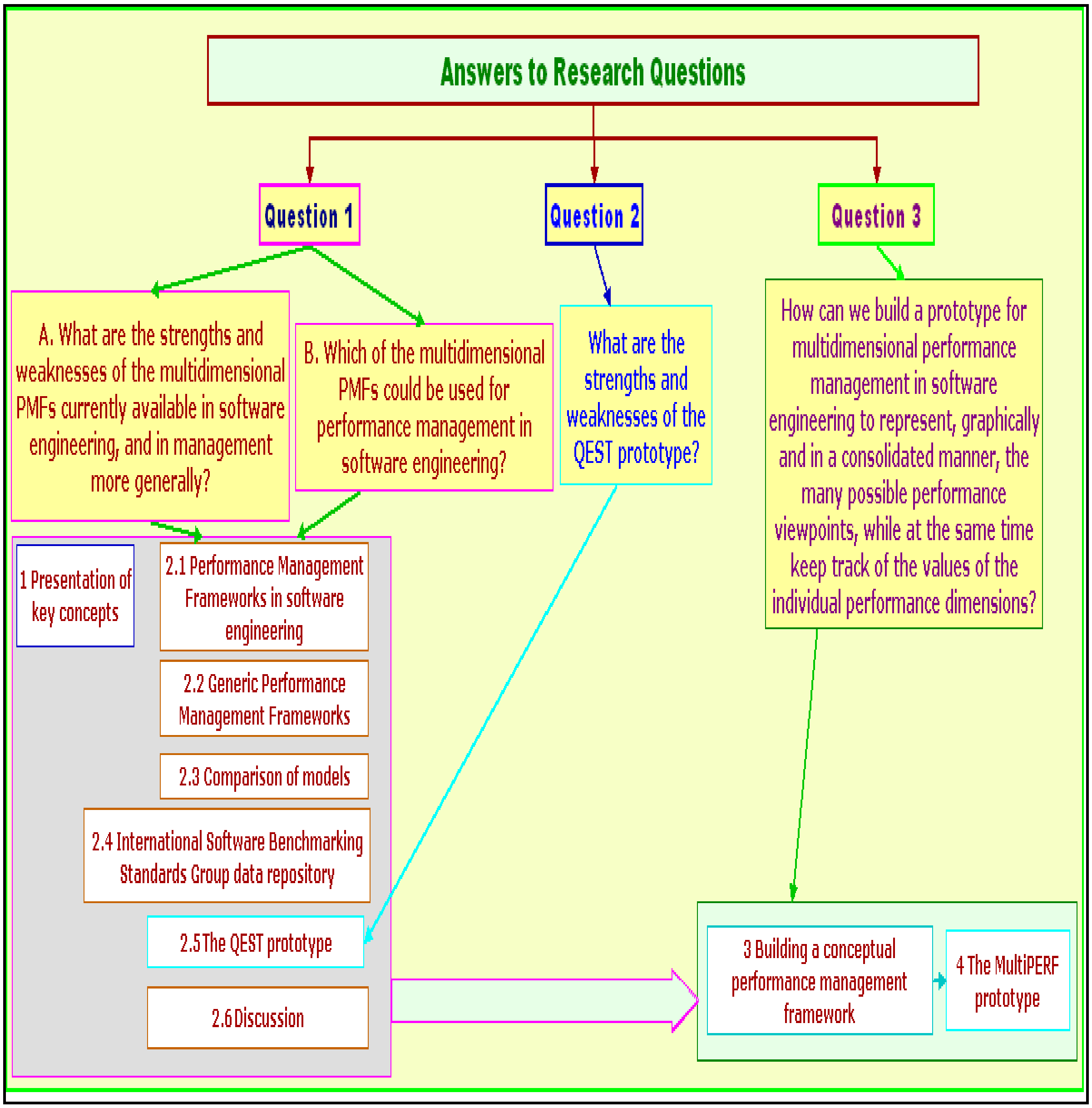


Figure-A II-1 Answers to research questions.

REFERENCES

12MANAGE. 2008. "Explanation of the DuPont Model". Accessed on (12.05.2008). <http://www.12manage.com/methods_dupont_model.html>.

Abran, A., P. Bourque, and I. Ndiaye. 2007. "Evaluation of a Black-Box Estimation Tool: A Case Study". Software Process: Improvement and Practice, vol. 12, n° 2, p. 199-218.

Abran, A., and L. Buglione. 2003. "A Multidimensional Performance Model for Consolidating Balanced Scorecards". Advances in Engineering Software, vol. 34, n° 6, p. 339-49.

Abran, A., J.-M. Desharnais, M. Maya, D. St-Pierre, and P. Bourque. 1998. *Design of Functional Size Measurement for Real-Time Software*. 13-23. Montreal: Université du Québec à Montréal. <http://www.gelog.etsmtl.ca/pub.html?category=Technical+Reports>. Accessed on (02.05.2008).

Abran, A., R. Dumke, J.-M. Desharnais, I. Ndiaye, and C. Kolbe. 2002. "A Strategy for a Credible&Auditable Estimation Process Using the ISBSG International Data Repository". IWSM-Magdeburg.

Abran, A., and J.-P. Jacquet. 1997. "From Software Metrics to Software Measurement Methods : A Process Model". Third IEEE International Software Engineering Standards Symposium and Forum – ISESS. Walnut Creek - Canada, (IEEE), p. 8.

Abran, A., M. Kunz, R. R. Dumke, and L. Buglione. 2003a. "A Prototype Web-Based Implementation of the QEST Model". IWSM 2003: Proceedings of the 15th International Workshop on Software Measurement Shaker Verlag, (Shaker Verlag GmbH), p. 82-92.

Abran, A., J. W. Moore, P. Bourque, and R. Dupuis (Eds). 2004. *Guide to Software Engineering Body of Knowledge*. Institute of Electrical and Electronics Engineers, 202 p.

Abran, A., A. Sellami, and W. Suryn. 2003b. "Metrology, Measurement and Metrics in Software Engineering". Software Metrics Symposium Ninth International. 3-5 September 2003, Sydney, Australia, p. 2-11.

Adams, C., and A. Neely. 2005. "Point of View Using the Performance Prism to Boost the Success of Mergers & Acquisitions". at (20.01.2008) <http://www.accenture.com/>.

Alonso, F., J. L. Fuertes, C. Montes, and R. J. Navajo. 1998. "A Quality Model: How to Improve the Object-Oriented Software Process". International Conference on Systems, Man, and Cybernetics. San Diego, CA, USA, (IEEE), (11-14 Oct.), p. 4884-9.

Anselmo, D., and H. Ledgard. 2003. "Measuring Productivity in the Software Industry". Communications of the ACM, vol. 46, n° 11, p. 121-125.

Artley, W., and S. Stroh. 2001. *The Performance-Based Management Handbook A Six-Volume Compilation of Techniques and Tools for Implementing the Government Performance and Results Act of 1993*. University of California. <http://www.orau.gov/pbm/pbmhandbook>. Accessed on (14.12.2008).

Balantzian, G. 2005. *Tableaux De Bord - Pour Diriger Dans Un Contexte Incertain*. France: Editions d'Organisation, 344 p.

Baldrige. 2005. "Baldrige Framework For Excellence". Accessed on (02.03.2007). <http://www.baldrige21.com/index.htm>.

Baldrige. 2010. "Criteria for Performance Excellence". Accessed on (10.04.2010). <http://www.baldrige.nist.gov/PDF_files/2009_2010_Business_Nonprofit_Criteria.pdf>.

Basili V.R., W. D. M. 1984. "A Methodology for Collecting Valid Software Engineering Data". IEEE Transactions on Software Engineering, p. 728-738.

Basili, V. R., G. Caldiera, and H. D. Rombach. 1994. "The Goal Question Metric Approach". Encyclopedia of Software Engineering, p. 528-532.

Bazzana, G., O. Andersen, and T. Jokela. 1993. "ISO 9126 and ISO 9000: Friends or Foes?". Software Engineering Standards Symposium. Brighton , UK (IEEE), p. 79-88.

Berlin, S., T. Raz, C. Glezer, and M. Zviran. 2009. "Comparison of Estimation Methods of Cost and Duration in It Projects". Information and Software Technology, vol. 51, n° 4, p. 738-748.

Berry, D. M. 1992. *Academic Legitimacy of the Software Engineering Discipline*. CMU/SEI-92-TR-34. Carnegie Mellon University: Software Engineering Institute, 79 p. Accessed on (28.11.2008).

Boehm, B. 1978. *Characteristics of Software Quality*. Amsterdam, Holland Elsevier, 210 p.

Boehm, B. W., J. R. Brown, and M. Lipow. 1976. "Quantitative Evaluation of Software Quality". 2nd International Conference on Software Engineering. San Francisco, California, United States, p. 592-605.

Bourne, M., M. Franco, and J. Wilkes. 2003. “Corporate Performance Management”. Measuring Business Excellence, vol. 7, n° 3, p. 15-21.

Bourne, M., J. Mills, M. Wilcox, A. Neely, and K. Platts. 2000. “Designing, Implementing and Updating Performance Measurement Systems”. International Journal of Operations & Production Management, vol. 20, n° 7, p. 754-771.

Bourque, P., R. Dupuis, A. Abran, W. J. Moore, and S. Wolff. 2002. “Fundamental Principles of Software Engineering – A Journey”. Journal of Systems and Software vol. 62, n° 1, p. 59-70.

Bourque, P., S. Oligny, A. Abran, and B. Fournier. 2007. “Developing Project Duration Models in Software Engineering”. Journal of Computer Science and Technology vol. 22, n° 3, p. 348-357.

Bourque, P., V. Stroian, and A. Abran. 2006. “Proposed Concepts for a Tool for Multidimensional Performance Modeling in Software Engineering Management”. IEEE-IES International Symposium on Industrial Electronics (ISIE). Montreal, (9-13 July 2006), p. 6.

British Quality Foundation. 2005. “The Excellence Model Framework”. Accessed on (07.07.2005). <http://www.quality-foundation.co.uk/#>.

Brown, M. G. 2001. *Baldrige Award Winning Quality: Eleventh Edition Covers the 2001 Award Criteria*. Productivity Press, 417 p.

Brown, M. G. 2003. *Baldrige Award Winning Quality: How to Interpret the Baldrige Criteria for Performance Excellence*, Twelfth Edition. Productivity Press, 338 p.

Brown , M. G. 2009. *The Pocket Guide to the Baldrige Award Criteria*, 15th Edition. Productivity Press, USA, 64 p.

Buglione, L., and A. Abran. 1998. “Multidimensional Software Performance Measurement Models: A Tetrahedron-based Design”. in 8th International Workshop on Software Measurement. Magdeburg, Germany, p. 12.

Buglione, L., and A. Abran. 1999. “Geometrical and Statistical Foundations of a Three-Dimensional Model of Software Performance”. Advances in Engineering Software, vol. 30, n° 12, p. 913-919.

Buglione, L., and A. Abran. 2001. “Multidimensionality in Software Performance Measurement: The QEST/LIME Models”. Proceedings of SSGRR (Second International Conference on Advances in Infrastructure for Electronic Business, p. 6.

Buglione, L., and A. Abran. 2002. “QEST nD: n-dimensional extension and generalisation of a software performance measurement model”. Advances in Engineering Software, vol. 33, n^{o} 1, p. 1-7.

Buglione, L., and C. Gencel. 2008. “Impact of Base Functional Component Types on Software Functional Size Based Effort Estimation”. In *Product-Focused Software Process Improvement*. Vol. 5089/2008, p. 75-89. Springer Berlin / Heidelberg.

Bunge, M. 1967. *Scientific Research. The Search for Truth*, 2. Berlin-Hidelberg- New York: Springer-Verlag, 374 p.

BusinessDictionary. 2005. “Online Business Dictionary”. Accessed on (30.10.2008). <http://www.businessdictionary.com/definition/performance.html>.

Cavano, J. P., and J. A. McCall. 1978. “A Framework for the Measurement of Software Quality”. ACM Software. Quality Assurance Workshop, p. 133-139.

Chatterjee, S., and A. S. Hadi. 2006. *Regression Analysis by Example*. Wiley-Interscience, 408 p.

Comstock, C., Z. Jiang, and P. Naudé. 2007. “Strategic Software Development: Productivity Comparisons of General Development Programs”. World Academy of Science, Engineering and Technology, n^{o} 34, p. 25-30.

Conti, T. A. 2007. “A History and Review of the European Quality Award Model”. The TQM Magazine, vol. 19, n^{o} 2, p. 112-128.

COSMIC. 2008. “The Common Software Measurement International Consortium”. Accessed on (15.09.2008). <http://www.cosmicon.com/>.

Crosby, P. B. 1995. *Quality without Tears: The Art of Hassle-Free Management*. McGraw-Hill Professional, 205 p.

Davies, J. 2008. “Integration: Is It the Key to Effective Implementation of the EFQM Excellence Model?”. International Journal of Quality & Reliability Management, vol. 25, n^{o} 4, p. 383-399.

DeMarco, T. 1982. *Controlling Software Projects: Management, Measurement, and Estimates*. Prentice Hall PTR, 296 p.

Déry, D., and A. Abran. 2005. “Investigation of the Effort Data Consistency in the ISBSG Repository”. 15th International Workshop on Software Measurement - IWSM'2005. Montreal, (Shaker-Verlag), p. 123-136.

Donaldson, S. E., and S. G. Siegel. 1998. "Measurement in Everyday Language". Accessed on (20.06.2008). <http://www.stsc.hill.af.mil/crosstalk/1998/02/measurement.asp>.

Donaldson, S. E., and S. G. Siegel. 2001. *Successful Software Development*, Second Edition. Prentice Hall, 729 p.

Dromey, R. G. 1995. "A Model for Software Product Quality". IEEE Trans. Softw. Eng, vol. 21, n° 2, p. 146-162.

Dromey, R. G. 1996. "Cornering the Chimera". IEEE Software, vol. 13, n° 1, p. 33-43.

Dykes, J., A. M. MacEachren, and M. J. Kraak. 2005. *Exploring Geovisualization*. Elsevier Ltd., 730 p.

Eccles, R. G., Jr. . 1991. "The Performance Measurement Manifesto". Harvard Business Review, vol. 69, n° 1, p. 131-137.

Education, O. R. I. f. S. a. 2010. "Guidelines for Performance Measurement". at (02.15.2010) <http://www.orau.gov/pbm/documents/g1201-5.pdf>.

Edvinsson, L. 1997. "Developing Intellectual Capital at Skandia". Long Range Planning, vol. 30, n° 3, p. 366-373.

Edvinsson, L. 2000. "Some Perspectives on Intangibles and Intellectual Capital 2000". Journal of Intellectual Capital, vol. 1, n° 1, p. 12-16.

Edvinsson, L., and M. S. Malone. 1997. *Intellectual Capital: Realizing Your Company's True Value by Finding Its Hidden Brainpower*. HarperBusiness, 240 p.

Encarta. 2008. "Dictionary". Accessed on (20.07.2008). <http://dictionary.msn.com/>.

English, R. 1991. "The Malcolm Baldrige Quality Award and Engineering Technology Education". Frontiers in Education Conference, 1991. Twenty-First Annual Conference. 'Engineering Education in a New World Order.' Proceedings., p. 30-33.

European Foundation for Quality Management. 2008. "EFQM". Accessed on (20.02.2008). <http://www.efqm.org/en/>.

Evans, J. R. 2004. "An Exploratory Study of Performance Measurement Systems and Relationships with Performance Results". Journal of Operations Management, vol. 22, p. 219-232.

Eveleens, J. L., and C. Verhoef. 2010. "The Rise and Fall of the Chaos Report Figures". Software, IEEE, vol. 27, n° 1.

Ewusi-Mensah, K. 2003. *Software Development Failures* The MIT Press 290 p.

Fay, R. 1995. *The Performance Imperative Strategies for Enhancing Workforce Effectiveness*. San Francisco: Paramount Publishing International Office, 456 p.

Fitzpatrick, R. 1996. "Software Quality:Definitions and Strategic Issues". Staffordshire University. Accessed on (20.01.2009). <http://www.comp.dit.ie/rfitzpatrick/papers/quality01.pdf>.

Flamholtz, E., and Y. Randle. 1998. *Changing the Game Organizational Transformations of the First, Second, and Third Kinds*. New York: Oxford University Press, 308 p.

Flynn, B. B., and B. Saladin. 2001. "Further Evidence on the Validity of the Theoretical Models Underlying the Baldrige Criteria". Journal of Operations Management, vol. 19, p. 617-652.

Galorath, D. D., and M. W. Evans. 2006. *Software Sizing, Estimation, and Risk Management: When Performance is Measured Performance Improves*. New York: Auerbach Publications, 574 p.

GAO. 2005. *Performance Measurement and Evaluation: Definitions and Relationships*. Coll. " United States Government Accountability Office Glossary "GAO-05-739SP. 5 p. <http://www.gao.gov/special.pubs/gg98026.pdf>. Accessed on (30.08.2008).

Gencel, C., L. Buglione, and A. Abran. 2009. "Improvement Opportunities and Suggestions for Benchmarking". In *Software Process and Product Measurement*. Vol. 5891/2009, p. 144-156. Springer Berlin / Heidelberg.

George, C., F. Cooper, and A. Douglas. 2003. "Implementing the EFQM Excellence Model in a Local Authority". Managerial Auditing Journal, vol. 18, n° 2, p. 122-127.

Ghalayini, A. M., and J. S. Noble. 1996. "The Changing Basis of Performance Measurement". International Journal of Operations & Production Management, vol. 16, n° 8, p. 63.

Giraud, F., O. Saulpic, G. Naulleau, and M.-H. Delmond. 2005. *Management Control and Performance Processes*, Gualino. 368 p.

Glass, R. L. 2005. "IT Failure Rates— 70% or 10–15%?". IEEE Software, vol. 22, n° 3, p. 110-111.

HCIL. 2010. "TreeMap". Accessed on (24.06.2010). <http://www.cs.umd.edu/hcil/treemap/>.

Hill , P. R., and C. Lokan. 2008. “The Benchmark Release 10”. Accessed on (30.11.2008), p. 103. <http://www.isbsg.org/ISBSGnew.nsf/WebPages/d6800b6f1b474377ca257476003f425a>.

HMC. 2000. *The American Heritage Dictionary of the English Language (AHD)*, 4. Houghton Mifflin Company, 2076 p.

Holloway, J., J. Lewis, and G. Mallory. 1995. *Performance Measurement and Evaluation*. Sage, 342 p.

Houldsworth, E., and D. Jirasinghe. 2006. *Managing and Measuring Employee Performance*. 245 p.

Howarth, P., and F. Redgrave. 2008. *Metrology - in Short* Third edition. Schultz Grafisk, DK 2620 Albertslund, 84 p.

IEEE Computer Society. 1990. *610.12-1990 IEEE Standard Glossary of Software Engineering Terminology* The Institute of Electrical and Electronics Engineers Inc. (IEEE).

IFPUG. 2008. “The International Function Point Users Group”. Accessed on (02.10.2008). <http://www.ifpug.org/>.

ISBSG. 2007. *Glossary of Terms V5.9.1.doc (25/09/2007)*. 19 p. Accessed on (20.05.2008).

ISBSG. 2008a. “The International Software Benchmarking Standards Group”. Accessed on (22.05.2008). <http://www.isbsg.org/>.

ISBSG. 2008b. “Software Development and Enhancement Project Data”. Accessed on (30.09.2008). <http://www.isbsg.org/>.

ISO. 1993. *International Vocabulary of Basic and General Terms in Metrology*. Geneva, Switzerland: International Organization for Standardization - ISO, 49 p.

ISO/IEC. 2001. *ISO/IEC 9126-1 Software Engineering - Product Quality - Part 1: Quality Model*. International Organization for Standardization, Geneva, Switzerland, 25 p.

ISO/IEC. 2002. *Systems and Software Engineering – Measurement Process 15939:2002*. International Organization for Standardization, Geneva, Switzerland, 37 p.

ISO/IEC. 2003a. “Software Engineering - COSMIC-FFP - a Functional Size Measurement Method ISO/IEC 19761:2003”.

ISO/IEC. 2003b. *Software Engineering - IFPUG 4.1 Unadjusted Functional Size Measurement Method - Counting Practices Manual ISO/IEC 20926:2003*.

Jørgensen, M., and K. Moløkken-Østvold. 2006. “How Large Are Software Cost Overruns? A Review of the 1994 Chaos Report”. Information and Software Technology, nº 48, p. 297-301.

Judson, A. S. 1990. *Making Strategy Happen Transforming Plans into Reality*. Basil Blackwell Inc., 250 p.

Jurgensen, R. C., M. P. Dolciani, and A. J. Donnelly. 1965. *Modern Geometry Structure and Method*. Boston: Houghton Mifflin Company, 592 p.

Kaplan, R., and D. Norton. 1992. “The Balanced Scorecard - Measures That Drive Performance”. Harvard Business Review vol. 70, nº 1, p. 71-79.

Kaplan, R., and D. Norton. 1993. “Putting the Balanced Scorecard to Work”. Harvard Business Review vol. 71, nº 5, p. 134-147.

Kaplan, R., and D. Norton. 1996a. *The Balanced Scorecard: Translating Strategy into action*. Harvard Business School Press, 303 p.

Kaplan, R., and D. Norton. 1996b. “Using the Balanced Scorecard as a Strategic Management System”. Harvard Business Review vol. 74, nº 1, p. 75-85.

Kemper, M., G. Rosso, and B. Monnone. 2006. *AdvancED Flash Interface Design*. Friends of Ed, 360 p.

Kerzner, H. 2009. *Project Management a Systems Approach to Planning, Scheduling, and Controlling*. Wiley, 1120 p.

Kilpi, T. 2001. “Implementing a Software Metrics Program at Nokia”. IEEE Software, vol. 18, nº 6, p. 72-77.

Kim, Y.-I. 2006. “Color and Symbolic Meaning of Elements in Nature”. Color Research and Application, vol. 31, p. 341-349.

Knowledge@Wharton. 2005. “Why So Many Big IT Investments Do So Little for Shareholder Value”. Wharton School Publishing. Accessed on (20.05.2008). <http://knowledge.wharton.upenn.edu/article.cfm?articleid=1228>.

KPMG. 2001. “Achieving Measurable Performance Improvement in a Changing World”. White Papers. Accessed on (01.09.2008), p. 32. <www.kpmg.com.au/aci/docs/measure-performance.pdf >.

Laitinen, E. K. 2002. “A Dynamic Performance Measurement System: Evidence from Small Finnish Technology Companies”. Scandinavian Journal of Management, vol. 18, p. 65-99.

Laitinen, E. K. 2009. “Importance of Performance Information in Managerial Work”. Industrial Management & Data Systems, vol. 109, n° 4, p. 550-569.

Laitinen, E. K., and G. Chong. 2006. “How do Small Companies Measure Their Performance?”. Problems and Perspectives in Management, vol. 4, n° 3, p. 49-68.

Laplante, P. A. 2004. *Software Engineering for Image Processing Systems* Boca Raton, Florida: CRC Press LLC, 220 p.

Lawton, G. 2009. “Users Take a Close Look at Visual Analytics”. Technology News, vol. 42, n° 2, p. 19-22.

Lebas, M. J. 1995. “Performance Measurement and Performance Management”. Int. J. Production Economics, vol. 41, n° 1-3, p. 23-35.

Li, D.-w., and W. Ke-yi. 2009. “Empirical Study on Factors Influencing on Corporate Market Value in It Industry Based on Skandia Navigator”. IEEE - 16th International Conference on Management Science and Engineering (ICMSE). Moscow, Russia, (Piscataway, NJ, USA), p. 488-495.

Liebesman, B. S. 1991. “The Malcolm Baldrige National Quality Award: A Prescription for Quality Improvement and International Competition”. Communications, 1991. ICC 91, Conference Record. IEEE International Conference on, p. 1138-1142 vol.2.

Lokan, C., and E. Mendes. 2009. “Applying Moving Windows to Software Effort Estimation”. 3rd International Symposium on Empirical Software Engineering and Measurement (ESEM). p. 111-122.

Lokan, C., T. Wright, P. R. Hill, and M. Stringer. 2001. “Organizational Benchmarking Using the ISBSG Data Repository”. IEEE SOFTWARE, p. 26-32.

Lynch, R. L., and K. F. Cross. 1995. *Measure Up: Yardsticks for Continuous Improvement* Blackwell Publishing. Oxford, 272 p.

Management, U. S. O. o. P. 2008. “Performance Management ”. Accessed on (20.09.2008). <http://www.opm.gov/perform/overview.asp>.

Marr, B., and A. Neely. 2003. *Automating Your Scorecard - The Balanced Scorecard Software Report*. CR-5102. Cranfield School of Management and Gartner, Inc, 322 p. Accessed on (04.05.2008).

Martinsons, M., R. Davison, and D. Tse. 1999. “The Balanced Scorecard : A Foundation for the Strategic Management of Information Systems”. Decision Support Systems, vol. 25, p. 71-78.

Mavroidis, V., S. Toliopoulou, and C. Agoritsas. 2007. “A Comparative Analysis and Review of National Quality Awards in Europe Development of Critical Success Factors”. The TQM Magazine, vol. 19, n° 5, p. 454-467.

McGarry, J., D. Card, C. Jones, B. Layman, E. Clark, J. Dean, and F. Hall. 2001. *Practical Software Measurement: Objective Information for Decision Makers*. 277 p.

McGraw-Hill's. 2008. “AccessScience Encyclopedia of Science & Technology Online”. Accessed on (29.10.2008). <http://www.accessscience.com/>.

Meridji, K. 2010. “Analysis of Software Engineering Principles from an Engineering Perspective”. PhD Thesis, École de technologie supérieure, 273 p.

Merriam-Webster. 2005. “Merriam-Webster Online”. Accessed on (30.05.2008). <www.merriam-webster.com/cgi-bin/mwwod.pl>.

Mohrman Jr., A. M., S. M. Resnick-West, and E. E. Lawler III (Eds). 1989. *Designing Performance Appraisal Systems Aligning Appraisals and Organizational Realities*. Coll. “ The Jossey-Bass Management Series ”. Jossey-Bass Publishers, 227 p.

Monitor, C. 1999. “Celemi's Annual Report”. at (20.02.2009) <http://www.sveiby.com/articles/CelemiMonitor99.htm>.

Morisawa, T. 2002. “Building Performance Measurement Systems with the Balanced Scorecard Approach”. Accessed on (01.09.2007), p. 17. <www.nri.co.jp/english/opinion/papers/2002/pdf/np200245.pdf >.

Morris, C. 2003. *Quantitative Approaches in Business Studies*, 6th edition Financial Times/ Prentice Hall, 512 p.

Neely, A. 2002. *Business Performance Measurement: Theory and Practice*. Cambridge (England): Cambridge University Press, 366 p.

Neely, A. 2005. “The Evolution of Performance Measurement Research: Developments in the Last Decade and a Research Agenda for the Next”. International Journal of Operations & Production Management, vol. 25, n° 12, p. 1264-1277.

Neely, A., and C. Adams. 2003. "The Performance Prism in Action – A Case Study". Control*,* vol. 29, nº 01, p. 14-17.

Neely, A., and C. Adams. 2005a. "Performance Prism". Encyclopedia of Social Measurement*,* vol. 3, p. 41-48.

Neely, A., and C. Adams. 2005b. "Perspectives on Performance: The Performance Prism". Accessed on (15.08.2008). <http://www.exinfm.com/pdffiles/prismarticle.pdf>.

Neely, A., C. Adams, and P. Crowe. 2001. "The Performance Prism in Practice". Measuring Business Excellence*,* vol. 5, nº 2, p. 6-12.

Neely, A., C. Adams, and M. Kennerly. 2002. *The Performance Prism the Scorecard for Measuring and Managing Business Success*. Great Britain: Pearson Education Limited, 393 p.

Neely, A., J. Mills, K. Platts, M. Gregory, and H. Richards. 1996. "Performance Measurement System Design: Should Process Based Approaches Be Adopted?". International Journal of Production Economics*,* vol. 46-47, nº 1, p. 423-431.

Niven, P. R. 2002. *Balanced Scorecard Step-by-Step: Maximizing Performance and Maintaining Results*. New York: John Wiley & Sons, Inc., 334 p.

OPM. 1996. "U.S. Office of Personnel Management - Archive". Accessed on (30.06.2010). <http://opm.gov/perform/articles/027.asp>.

Park, R. E., W. B. Goethert, and W. A. Florac. 1996. *Goal-Driven Software Measurement — a Guidebook*. Handbook CMU/SEI-96-HB-002. Software Engineering Institute Carnegie Mellon University, 189 p. Accessed on (02.02.2007).

Paulson, D., and S. Zahir. 1995. "Consequences of Uncertainty in the Analytic Hierarchy Process: A Simulation Approach". European Journal of Operational Research*,* vol. 87, nº 1, p. 45-56.

PMI. 2000. *A Guide to the Project Management Body of Knowledge* Third Edition. Project Management Institute, 209 p.

Porter, L. J., and S. J. Tanner. 2004. *Assessing Business Excellence*, Second Edition. Elsevier Butterworth-Heinemann, 468 p.

Posey, T. 2006. "Whatever Happened to the DuPont Formula?". Accessed on (30.05.2008). <http://www.crystalequityresearch.com/>.

Pressman, R. 2001. *Software Engineering - a Practitioner's Approach*, 5. Boston, Mass. : McGraw Hill, 860 p.

Ramesh, R. 2000. *Financial Analyst's Indispensable Pocket Guide* McGraw-Hill, 218 p.

Reo, D., N. Quintano, and L. Buglione. 1999. *ESI Balanced It Scorecard Infrastructure&Innovation Perspective, Technical Report*. ESI-1999-TR-043. European Software Institute. Accessed on (21.02.2007).

Rini, V. S., and B. Egon. 1999. *Goal/Question/Metric Method: A Practical Guide for Quality Improvement of Software Development*. McGraw-Hill Inc., US, 280 p.

Robbins, S. P. 2007. *The Truth About Managing People*, Second Edition. FT Press, 240 p.

Rummler, G. A., and A. P. Brache. 1991. *Improving Performance How to Manage the White Space on the Organization Chart*. Jossey-Bass Publishers, 227 p.

Saaty, T. L. 1990. "How to Make a Decision: The Analytic Hierarchy Process". European Journal of Operational Research, vol. 48, n° 1, p. 9-26.

SAIC. 2009. "SAIC from Science to Solutions". Accessed on (20.02.2009). <http://www.saic.com/>.

Schmietendorf, A., A. Scholz, and C. Rautenstrauch. 2000. "Evaluating the Performance Engineering Process". Proceedings Second International Workshop on Software and Performance WOSP 2000. Ottawa, Ontario, Canada (Association for Computing Machinery), (17-20 September), p. 89-95.

Scorecard, B. 2009. "Hall of Fame". Accessed on (14.02.2009). <http://www.thepalladiumgroup.com/ABOUT/HOF/Pages/overview.aspx>.

Shapiro, F. R. 2000. "Origin of the Term Software: Evidence from the JSTOR Electronic Journal Archive". IEEE Annals of the History of Computing vol. 22, n° 2, p. 69-71.

Simons, R. 1995. *Levers of Control: How Managers Use Innovative Control Systems to Drive Strategic Renewal*. Boston, Massachusetts: Harvard Business School Press, 217 p.

Sink, D. S. 1985. *Productivity Management: Planning, Measurement and Evaluation, Control and Improvement*. New York & Toronto: John Wiley & Sons Inc, 518 p.

Sink, D. S., and T. C. Tuttle. 1989. *Planning and Measurement in Your Organization of the Future*. Norcross: Engineering & Management Press, 331 p.

Skandia. 1994. "Visualizing Intellectual Capital in Skandia". Supplement to Skandia's 1994 Anual Report. Accessed on (30.04.2010), p. 24. <http://www.skandia.com/financials/pdfs/e9412Visualizing.pdf.>.

Skandia. 2006. “Skandia Insurance Company Ltd”. at (31.05.2010) <http://www.skandia.com/en/index/>.

Smith, S. A., C. W. Nelson, R. K. Koss, M. L. Keedy, and M. L. Bittinger. 1992. *Informal Geometry*. Addison Wesley Publishing Company 654 p.

Sokolowski, J. A., and C. M. Banks. 2009. *Principles of Modeling and Simulation: A Multidisciplinary Approach* Wiley, 280 p.

Srivivatanakul, T., and B. H. Kleiner. 1996. “Developing a plan to win the Baldrige Award”. Training for Quality*,* vol. 4, n^{o} 4, p. 12-15.

Stroian, V. 1999. “Un Environnement Automatisé Pour Un Processus Transparent D'estimation Fondé Sur La Base De Données Du International Software Benchmarking Standards Group (ISBSG)”. Master's degree thesis, Montreal, Université du Québec à Montréal, 47 p.

Stroian, V., P. Bourque, and A. Abran. 2006. “Visualization - a Key Concept for Multidimensional Performance Modeling in Software Engineering Management”. IEEE-TTTC International Conference on Automation, Quality & Testing, Robotics AQTR 2006 (THETA 15). Cluj Romania, (25-28 May), p. 6.

Stylianou, A., C., and R. L. Kumar 2000. “An Integrative Framework for IS Quality Management”. Commun. ACM*,* vol. 43 n^{o} 9 p. 99-104

Sveiby, K.-E. 1997a. “The Intangible Assets Monitor”. Accessed on (02.10.2007). <http://www.sveiby.com/articles/IntangAss/CompanyMonitor.html>.

Sveiby, K.-E. 1997b. *The New Organizational Wealth Managing & Measuring Knowledge-Based Assets*. San Francisco: BK Berrett-Koehler Publishers,Inc, 220 p.

Sveiby, K.-E. 2001. “What is Knowledge Management?”. Accessed on (02.02.2009). <http://www.sveiby.com/articles/KnowledgeManagement.html>.

Sveiby, K.-E. 2004. “Methods for Measuring Intangible Assets”. *www.sveiby.com*. at (10.10.2008)

Taylor, A. 2000. “IT Projects Sink or Swim”. The Computer Bulletin*,* vol. 42, n^{o} 1, p. 24-26.

The Standish Group. 1995. “Chaos”. Accessed on (30.05.2008), p. 8. <http://www.standishgroup.com/>.

The Standish Group. 2009. “Chaos Summary 2009 the 10 Laws of Chaos”. Accessed on (01.07.2010), p. 4. <http://www.standishgroup.com/>.

TRW. 2009. “TRW Incorporated”. Accessed on (27.02.2009). <http://en.wikipedia.org/wiki/TRW>.

Tudahl, D., and R. Lindner. 1994. “Software Development at a Baldrige Winner”. Electro/94 International. Conference Proceedings. Boston, MA , USA p. 167-180.

VIM. 2008. *International Vocabulary of Metrology — Basic and General Concepts and Associated Terms (VIM)*. BIPM (Bureau International des Poids et Mesures) 3rd edition, JCGM 200:2008 90 p. <http://www.bipm.org/utils/common/documents/jcgm/JCGM_200_2008.pdf>.

Vincenti, W. G. 1990. *What Engineers Know and How They Know It: Analytical Studies from Aeronautical History* The Johns Hopkins University Press, 336 p.

Vliet, H. v. 2008. *Software Engineering: Principles and Practice* Third Edition. Glasgow, Great Britain: John Wiley & Sons, 740 p.

Weisstein, E. W. 2002. *CRC Concise Encyclopedia of Mathematics*, Second Edition. Chapman & Hall/CRC, 3252 p.

Westerveld, E. 2003. “The Project Excellence Model1: Linking Success Criteria and Critical Success Factors”. International Journal of Project Management, vol. 6, n^{o} 6, p. 411-418.

WordReference. 2005. “Multidimensional”. Accessed on (25.05.2008). <http://www.wordreference.com/definition/multidimensional>.