

# A Model for Selecting an Optimum Set of Measures in Software Organizations

Ansar Malook Bhatti<sup>1</sup>, Hafiz Muhammad Abdullah<sup>1</sup>, Cigdem Gencel<sup>1</sup>,

<sup>1</sup> Blekinge Institute of Technology, School of Engineering,  
372 25 Ronneby, Sweden  
{ambh07, hmabh07}@student.bth.se, cigdem.gencel@bth.se

**Abstract.** Most of the software organizations face difficulties in choosing the measures to collect since there is no universal set of measures for all types of organizations and projects. Experience shows that measurement can be more successful if the measures are collected based on the goals of the organization or the project which it will serve. However, one of the major constraints for the organizations is the associated cost for the resources needed when collecting the measures. Therefore, based on their goals, the software organizations require collecting not only as few measures from a large number of possible measures as possible but an optimum set of measures as well. In this paper, we propose a model, called 'Optimum Measures Set Decision (OMSD) Model', which is an extension of the well-known Goal Question Metric (GQM) paradigm using a heuristics approach. We performed a survey by distributing a structured questionnaire to a number of people from the industry in order to evaluate and get feedback on these factors. We evaluated the rules of the model by means of some sample cases we created. In this paper, we discuss OMSD as well as the empirical studies we conducted in order to develop it.

**Keywords:** Software Measurement Program, Software Measures, Software Process Improvement, Goal Question Metric.

## 1 Introduction

Software measurement process has become an integral part for software process due to its significance in project estimations, decision making and software process improvement [1]. However, in spite of the fact that many organizations started measurement programs to benefit from it; the failure rates for software measurement programs in software organizations are still very high.

About 80% of the measurement programs were reported to fail to either helping in decision making or delivering performance improvements for numerous reasons [2], [3]. Some of the most significant reasons stated are as follows [2], [4], [5]:

- Focusing on collecting process rather than having clear action plans for improving the organizational processes and/or making decisions,
- Inappropriate measures selection; a misunderstanding of what is to be measured, why and how it is to be measured,

- Inadequate data collection and wrong interpretations of data that leads to ineffective decision making,
- Lack of trained and expert resources required to dedicate to measurement,
- Lack of management support for the measurement program,
- The cost for measurement not planned according to the organization's budget.

Various frameworks and models have been developed to overcome some of the above mentioned difficulties software organizations are facing, such as Goal Question Metric (GQM) paradigm [6], [7], Goal Question Indicator Model (GQIM) [8] and Measurement Information Model [9]. GQM; developed by Basili and Weiss [6] and then improved by Basili and Rombach [7], is one of the well-known frameworks used in deriving measures from organization or business goals. Two reasons for the success of GQM are stated in [10] as that it is adaptable to many different organizations and environments and it aligns with the organizational directions and goals.

However, although these frameworks help the organizations to collect data on the measures which are required to fulfill the goals of the organization, none of those explicitly support the need to limit the number of measures to be collected [10]. In fact, one of the major constraints for the organizations which are also one of the significant reasons for measurement programs failure is the associated cost for the measurement programs.

A well-known figure, Tom De Marco said [11]; *"Metrics are good, more would be better and most is best but the importance of cost and time factor cannot be denied. Faced with a high number of measures to be collected for software process improvement reasons, most organizations want to know whether all those measures are equally important or some are more important than the others"*. Two out of ten problems leading to failure in the implementation of software measurement programs are reported by Howard Rubin to be: the intensive use of a single measure or, conversely, the use of too many [12].

According to [13], software measurement programs usually fail as they require expert judgment for selecting appropriate number of measures in relation to the organizational goals. The mapping of goals with appropriate measures requires experienced resources in the field of software measurement. These goals are required to be prioritized. One important point to be considered is that this prioritization might also be influenced by the cost associated to measures collection. Therefore, software organizations require deciding on an optimum set of measures which are good enough and at the same time less costly.

This paper suggests a model named 'Optimum Measure Set Decision (OMSD) Model' which extends the GQM approach and aims to fill in the gap discussed above by facilitating the managers in selecting an optimum set of measures from a large number of possible measures. To develop the model, we identified the factors which are significant when deciding on the measures to be collected as well as optimizing the cost associated based on the findings of an extensive literature review and getting feedback from the industry by conducting a survey. Then, we tested the model by means of some sample cases we created.

The paper is divided into five main sections. Section 1 provides an introduction. Section 2 explains the proposed OMSD model. Section 3 presents the empirical

studies we made in order to test the model and discusses the results we obtained. Finally, we present the conclusions in Section 5.

## **2 Optimum Measures Set Decision (OMSD) Model**

The Optimum Measures Set Decision (OMSD) Model [14], which is extending the GQM approach, is based on a heuristics approach. Heuristics is defined as a technique which seeks near optimal solution at a reasonable cost [15]. It is a rather flexible, easy to understand and implement technique. Constraints [16] regarding the costs and resources are defined early in the measurement process and it plays an important role during the final decision making on an optimum measures set selection. These constraints act as thresholds which are utilized as process terminators in OMSD Model.

The constraints for the heuristics rules are collected after the first level of GQM is implemented; i.e. when the goals are identified. After implementing levels in GQM, all the measures decided are ensured to be collected for a purpose and hence also reflect interesting and useful measurements for an organization. OMSD consists of five main steps shown as follows (Fig. 1 below):

- Category Selection
- Attributes Identification
- Measures Selection
- Collecting Data on the Measures Based on Factors
- Decision Making

### **2.1 Category Selection**

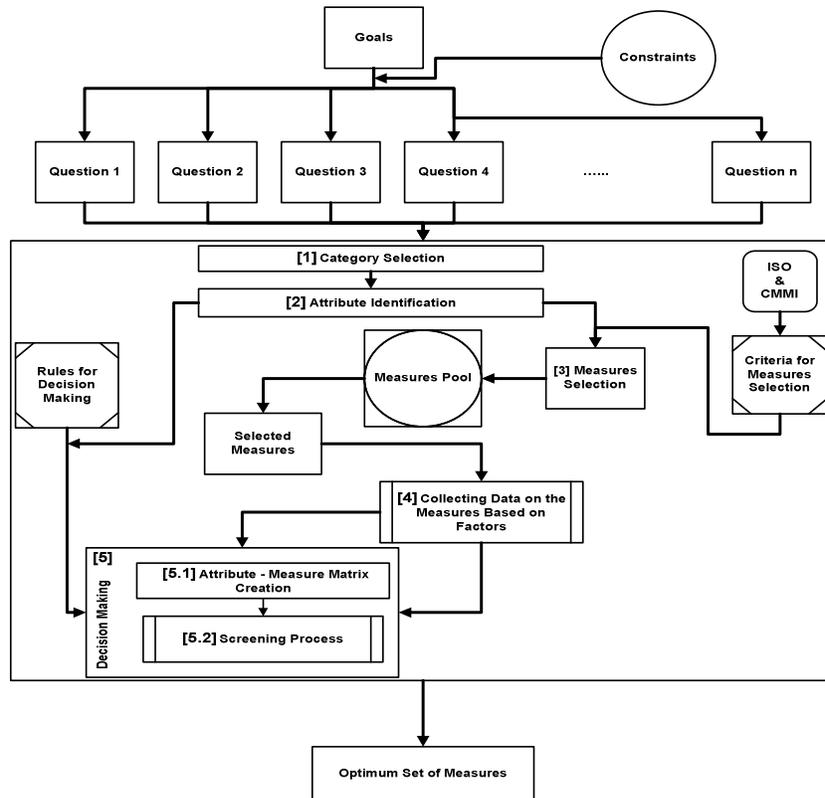
In order to perform any measurement activity we need to identify the entity to be measured and the associated attributes [17]. This step involves mapping of the questions identified in the questions level of GQM paradigm on their respective entity categories. In [17], three main categories of entities are defined as: Process, Product and Resource.

Process category includes different activities and these activities are associated with a timescale. There is a particular order defined for these activities which means activity B requires the completion of activity A. This timing could be implicit or explicit. Resources and Product categories are associated with the process category. Every process has certain resources and products that it utilizes. This step results in the identification of measurement entities (questions) on their respective classes which serves as input to the next phase of 'Attribute Identification'.

### **2.2 Attribute Identification**

Attributes associated with the entities are identified that can be divided into two main categories as external and internal attributes [17]. Internal attributes are those which

could be measured only by observing the product. External attributes include processes, products, resources and its behavior which tells how these attributes relate to the environment. Category selection and attributes identification provide deep understanding regarding behavior of the respective questions.



**Fig. 1.** Optimum Measures Set Decision (OMSD) Model

This step results in the following two outputs regarding the questions:

- Respective categories of questions
- Associated attributes depicting their behavior and relation with environment.

These identified attributes facilitate in Decision Making (Step 5) later on. At least one attribute is identified for each question because these attributes represent its respective questions in decision making. It is possible that one question can be related to more than one attribute and one attribute can be associated with more than one question. These dependencies are also identified and used later in the decision making process.

### 2.3 Measures Selection

The main aim of this step is the selection of all possible measures from a ‘Measures Pool’ using identified categories attributes and measures selection criteria (explained below). This step is conducted at the third level of GQM when the measures are identified.

We define the ‘Measures Pool’ as the repository which contains a finite set of measures defined for the attributes of software entities. Since this set might contain hundreds of measures, as our first attempt in this study, we included the ones provided in ISO 9126 [18], [19], [20], [21], ISO 25000 [22], the ones suggested as the minimum set of measures defined in CMMI [23] and the measures which are best known by the organization itself in their experience factory. Every organization can tailor the attributes and measures set in the Measures Pool based on their needs.

We defined these criteria for the measures selection from the Measures Pool based on the guidelines defined by the standards such as ISO 15939 [9] and CMMI [23]. We incorporated the following criteria in our model:

- Feasibility of collecting data in an organization
- Availability of human resources to collect and manage data
- Extent of intrusion and disruption of staff activities
- Availability of appropriate tools and equipment
- Personal preference
- Ease of interpretation by measurement users and measurement analysts
- Ease of presentation and relevancy to the audience

The criteria serve as a base for measures selection, but these do not tell which of the available measures should be collected for a specific attribute. For example, if a measure for software size is required to be collected, all size measures available in the Measures Pool such as Function Points, SLOC, Bytes, are selected.

### 2.4 Collecting Data on the Measures with respect to the Identified Factors

Once the measures are selected based on the basic criteria defined in the previous section, the factors (Fig. 2) which we identified to be significant in deciding on the optimum set are considered for further decision.

A number of factors have been suggested to be considered when selecting the measures such as cost, time, resource requirements, tools, special trainings etc. [24], [25], [11]. We have selected the most basic ones having significant impact on the measurement process. These factors are general and can be applied to any process that involves measurement irrespective of its domain i.e. software process, management process, manufacturing process etc.

Factors proposed in the OMSD Model are the core of the model and play a key role in getting important data for selected measures which is vital for deciding on the optimum measures set. By means of the survey<sup>1</sup>, we received industrial feedback on the identified factors. Survey was conducted in 10 different software organizations

---

<sup>1</sup> For the questionnaire design, see <http://sites.google.com/site/omsd09/survey>

from different countries and which have different maturity levels. Respondents are selected on the basis of their experience regarding software measurement activities in order to create better understanding of our defined factors and having reliable feedback from them.

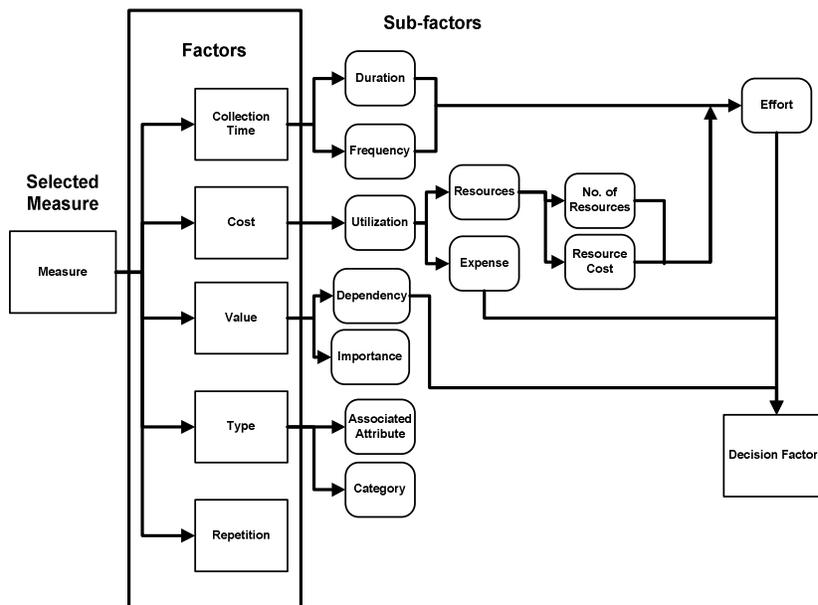
In this step, for each measure, the relevant data for each factor are to be entered by the measurement responsible and used in final decision making.

**Factor 1: Collection Time-** The collection time for a measure is composed of two sub-factors which are Duration and Frequency.

Duration describes the time required to collect a measure 'A'. Frequency describes how many times that specific measure 'A' is needed to be collected. Based on these data, the cumulative collection time weight (CCTW) is calculated by the following formula;

$$CCTW = Duration \times Frequency \quad (1)$$

**Factor 2: Cost-** The cost of a measure is determined based on the resources required (both human and non-human). It is comprised of one sub-factor: Utilization. Utilization factor is related to two sub-factors which are 'Resources' and 'Expense'. Resources contain the details regarding the number of resources required as well as their cost as; No of Resources and Resource Cost. Expense involves any other additional expenses such as the requirements for training, tools, hardware, etc.



**Fig. 2.** Significant Factors for Measures Selection

We defined three classes of resources with respect to their roles in an organization as Upper Management Resource (UMR), Middle Management Resource (MMR) and

Resource [Developer, Tester, Analyst] (R). Individual resource cost (IRC) is calculated by the following formula;

$$IRC = \frac{Salary}{Working\ Hours\ per\ Month} \quad (2)$$

Resource cost is used with the cumulative time weight in calculating effort of a particular measure. It is calculated by adding the cost of different resources involved in it.

$$Resource\ Cost\ (RC) = \sum_{i=1}^n IRC_i \quad (3)$$

In these calculations, we assumed that the working hours in a month depend on the organizational work policy. Currently we assumed that resources work 8 hours daily and 22 days per month means they work total 176 hours per month.

**Factor 3: Value-** Value is defined as a measure's dependency to other measures (direct and derived measures [26]) and its importance in a client organization's view as its Dependency and its Importance. This dependency identification is critical for effectiveness of final decision making. In OMSD, we used only direct measures in order to make decision process easier. If a derived measure came up we adjust it by calculating the cost of each base measure and then add them together. For example, cost of 'Development effort' measure is calculated by adding cost of measuring 'number of persons' and 'number of hours'. By means of that, we avoided complexity of dependency between measures. On the other hand, Importance of a measure (it is related to the priority of the goals identified) depicts the significance of a particular measure in the view of customer. We have defined four levels of importance which includes Level 1: Minor, Level 2: Essential, Level 3: Major, Level 4: Critical.

**Factor 4: Type-** The type of a measure is collected for information purposes only. The type is defined as associated attributes and Category.

**Factor 5: Repetition-** Data on this factor is also collected only for information purposes. It facilitates in decision making later on by identifying the multiple usage/repetition of the same measure. In this way, it reduces the probability of redundancy.

**Decision Factors:** Decision factors include the effort required to collect a particular measure.

Effort is calculated in terms of person-hours through cumulative collection time weight and resource cost as;

$$Effort = CCTW \times RC \quad (4)$$

Then, the effort is used to calculate the cumulative cost (CC) for collecting a particular measure. And it is calculated by the following formula; (see also Factor 2: Cost).

$$CC = Effort + Expense \quad (5)$$

## 2.5 Decision Making

Decision making is the final step in the OMSD model. Decision making is a process that shows expertise of an individual in selecting one solution out of the possible alternatives [27]. There are different categories given in the literature [27] that differentiate different decision making problems.

Decision making in OMSD model includes controlled inputs in the form of constraints (time and cost limits) and variables as factors such as usage and importance. This cost limit is used as Constraint cost (Ccost) while executing 7 steps of the screening process (explained below). So, the problem addressed by OMSD model is deterministic and decisions are made by constructing rules (7 steps explained below) in order to solve the defined problem. The Decision Making step consists of Attribute – Measure Matrix Creation and Screening Process. Main purpose of this step is to decide on an optimum set of measures from the selected measures. It utilizes the identified attributes (Step 2 of OMSD) and selected measures (Step 3 of OMSD) for decision making. Some ground rules are defined which facilitate in the final decision making.

**Attribute – Measure Matrix Creation:** This is first step for decision making and identified attributes and the selected measures are its inputs. It is a two dimensional matrix that depicts the relation of the measures with their respective attributes (see Fig. 3 below).

Measure Attributes	Measure 1	Measure 2	Measure 3	.....	Measure n
Attribute 1	0	0	1	0	1
Attribute 2	1	0	0	0	0
Attribute 3	0	1	0	0	0
...	1	0	0	0	0
...	0	0	0	1	0
...	1	0	0	0	1
Attribute n	0	0	0	1	0
<b>Usage</b>	1	1	1	1	1
<b>Importance</b>	4	4	3	3	2
<b>Cumulative Cost</b>	50 €	70 €	10 €	15 €	40 €

Fig. 3. Attribute – Measure Matrix

First, the dependencies of the measures are identified. The reason for this is that OMSD in this current form cannot handle dependencies between the base and the derived measures. Therefore, the model might exclude a particular measure in the screening process, on which many measures could be dependent. Therefore, first consider only direct measures in this step and when any derived measure comes up, instead of using its base measures, we use the derived measure itself.

Attribute – Measure Matrix creation consist of the following steps:

1. Sort measures on the basis of their importance. Highest importance (4-Critical) measure comes first.
2. If Measure 'X' is used to measure Attribute 'Y', then fill in the respective cell with '1' otherwise with '0' (see Fig. 3 above).
3. Calculate the number of usages of each measure by adding the values in that column.
4. Add Importance weight and cumulative cost of each measure in the respective cells.

**Screening Process:** Screening process is the last step in OMSD Model. It utilizes attribute – measure matrix and a set of pre-defined decision factors in order to decide on the optimum set of measures. The defined screening rules are implemented during the final screening process. These rules are based on a number of factors such as the Number of usage, Importance and Cost. A heuristics approach is used to make a tradeoff. After every selection, a comparison is made with the constraints (cost and time) [16] in order to control the progress. Certain tradeoffs are also needed with respect to the importance of the measure and the cost for the measure, but these tradeoffs are primarily dependent on the organizational business needs and priorities.

The model selects at least one measure for each attribute because each attribute represents a particular question. Order of these rules could be changed based on the organizational requirements. Steps include in screening process are given below:

**Step: 1** Select Attribute 'Ai' Where  $i = \{1,2,3,4,\dots,n\}$

**Step: 2** Select each measure 'mi' which satisfy attribute 'Ai'

$A_i = \{m_1, m_2, m_3, \dots, m_n\}$

**Step: 3** Calculate the usage  $U_{mi}$  of each measure  $m_i$

Usage of  $m_i$  ( $U_{mi}$ )=How many time it is used in Attribute-Measure Matrix.

**Step: 4** Perform Comparisons on the base of Decision factors.

- Compare the Use of each measure  $m_i$  with all selected measures
- Compare the Importance of each measure with all selected measures  
*Importance = weight assigned to Measure  $m_i$  (Step 4: Factor-Value)*
- Compare the Cost of each measure with all selected measures  
*Cost = Calculated through Step 4 of the Model*

**Step: 5** Measures are selected after step 4. Note that measure selection is completely dependent on the organizational decision (means which decision factor is of high importance for organization).

**Step: 6** Check the selected measure against the pre-defined constraints. Primary aim of this step is to control the measurement process in order to make sure that cost of the selected measures remains under the cost limits. This step is repeated at the end of each iteration during the screening process.

**Step: 7** Check the following conditions:

- If Cost of selected measure ( $C_{ms}$ ) is less than Constraint Cost ( $C_{cost}$ ) then continue from step 1

$$\text{Cms} < \text{Ccost}$$

(6)

- If  $\text{Cms} > \text{Ccost}$  terminate the process.

Step 6 and 7 are mutually exclusive. Once the selected measure is analyzed against the defined constraints, decision about continuation or termination of measurement process is made on the basis of condition in step 7. An optimum measure set is selected after completion of these steps (either termination condition is met or all measures are gone through).

*A Sample Case.* In order to evaluate this model, we created a sample case<sup>2</sup> and executed it to obtain the optimum set of measures for a specified set of goals. Specifically, 5 goals were defined and 11 questions were identified which would provide the information required. Each question can be answered by means of a number of measures associated with it. For this case, we identified 23 suitable measures from the Measures Pool.

Then, we collected the data for each measure according to the factors defined in OMSD. Among those measures, we observed that 3 of the measures are to be used for answering more than one question. Therefore, we had 20 measures for the further decision making process. After this step, we calculated the effort required to collect each measure, checked the dependencies of the measures, and at the end calculated the cumulative cost for each measure by adding any additional expenses if exist. We performed the same calculations for the all 22 measures.

Next, we created the 'Attribute-measure matrix' and then made the decisions using the screening rules in the OMSD model. In this matrix, cumulative costs for all of the measures are obtained in the previous step along with the Importance value decided by the managers according to the importance of each goal. For example, 'Productivity' measure had a cumulative cost 350\$ and its importance value is 3. These measures are then mapped to the relevant attributes. For example, Development effort measure as well as any product size measure is required to derive the measure for answering the question related to the productivity attribute. Using the screening rules, we saw that two size measures can be chosen to derive the measure for the productivity. Therefore, we first considered all three measures (Development Effort, SLOC, FP). Then, we checked the 'No. of usage' attribute for each measure globally. FP is to be used more than SLOC, which means that it can be used to satisfy other goals as well. Here, SLOC and FP are equally important and the cost for measuring FP is higher. Based on this information, the OMSD model decided to choose FP since it can be used to answer a couple of questions which reduces the total cost.

After each execution for each measure, there checked the Constraint cost. After selection of each measure, the remaining available budget is re-calculated by deducting the cost for the selected measure from it. It is important to note that if two measures have the same 'No. of usage' but different importance and cost, then a tradeoff between importance and cost is made by the measurement responsible(s) and/or managers.

---

<sup>2</sup> The sample case can be found at: <http://sites.google.com/site/omsd09/sample-case>

At the end of the whole process, OMSD model decided on 8 measures from suitable 23 measures. This is the optimal set of measures as it helps achieving the goals under the defined constraints and identified factors.

Although we obtained a smaller measures set in this experiment, our main purpose in this case study was not to show the model's efficiency but rather to test the applicability and the rules of the model. The model is dependent on the initial measures set as well as the constraints set by the measurement responsible.

## 4 Conclusions

Measurement process is one of the critical processes, which leads organizations towards process improvement. Since numbers of measures are available, it is needed to have an approach guiding organizations for selecting an optimum set of measures. This paper presented a model called "OMSD Model", a systematic approach for dealing with the challenge of 'finding an optimum measures set' out of the possibly large set of measures.

In a nutshell, this model is developed to address one of the challenges organizations are facing; the risk for the failure of measurement programs due to improper time and cost estimates, by minimizing the cost by supporting efficient and effective measures selection process in organizations. There is little explicit discussion in the literature about what constitutes a reasonable overhead for a measurement program. In [6], it is stated that 90% of the practitioners reported to spend less than 3% of their time on metrics-related work. OMSD model can also help the organizations to collect such information so that they can also calculate the Return on Investment (ROI) for initiating such programs.

Even though we evaluated the OMSD model by means of an industrial survey we made to determine the factors considered in the model and by means of a thorough experimentation of the rules of the model, in order to show evidences that the model is valuable for the organizations, industrial case studies should be conducted.

One of the current constraints of the OMSD Model is that high levels of human interaction are needed to enter the input required by the model such as measures dependency, time and cost limits. Improvement and automation of this process can reduce human effort resulting in less time and cost expense.

Other future works related to this study includes measures prioritization based on the priority of the goals at Step 1 of GQM, developing a Measures pool that will make initial measures selection easier (Step 3 of OMSD Model), incursion of new factors based on more industrial surveys, industrial experimentation of the OMSD Model and its integration with measurement frameworks other than GQM.

**Acknowledgements.** We would like to thank Johan Holmgren for his support in the development of the heuristics approach. We also thank the survey respondents for their feedback on the model.

## References

1. Wang, Q. and Li, M.: Measuring and improving software process in China. International Symposium on Empirical Software Engineering, 10 (2005)
2. Bundschuh, M., and Dekkers, C.: The Measurement Compendium: Estimating and Benchmarking Success with Functional Size Measurement. Springer-Verlag, Berlin Heidelberg (2008)
3. Goethert, W., Hayes, W.: Experiences in Implementing Measurement Programs. Technical Note, Software Engineering Institute, Carnegie Mellon University, CMU/SEI-2001-TN-026 (2001)
4. Gopal, A., Krishnan, M.S, Mukhopadhyay, T., Goldenson, D.R.: Measurement Programs in Software Development: Determinants of Success. IEEE Transactions on Software Engineering, Vol.28, Issue 9, pp. 863--875 (2002)
5. Hall, T., and Fenton, N.: Implementing Effective Software Metrics Programs. IEEE Software, Vol.14, Issue 2, pp. 55--65 (1997)
6. Basili, V.R, and Weiss, D.M.: A Methodology for Collecting Valid Software Engineering Data. IEEE Transactions on Software Engineering, Vol.SE-10, no.6, pp. 728--738 (1984)
7. Basili, V.R, and Rombach, H.D.: The TAME Project: Towards Improvement-Oriented Software Environments. IEEE Transactions on Software Engineering, SE-14 No.6, pp. 758--773 (1988)
8. Park, R.E., Goethert, W.B., Florac, W.A.: Goal-Driven Software Measurement - A Guidebook, Handbook. Software Engineering Institute, Carnegie Mellon University, CMU/SEI-96-HB-002 (1996)
9. ISO/IEC 15939 International Standard 1<sup>st</sup> Edition 2002: Software engineering — Software measurement process. Reference Number ISO/IEC 15939:2002(E) (2002)
10. Berander, P. and Jönsson, P.: A goal question metric based approach for efficient measurement framework definition. In: Proceedings of the 2006 ACM/IEEE international Symposium on Empirical Software Engineering (Rio de Janeiro, Brazil, September 21 - 22, 2006). ISESE '06. ACM, pp. 316--325 New York (2006)
11. DeMarco, T.: Why does software cost so much? Essay 2: Mad about Measurement. Dorset House Publishing, Co., Inc., New York, NY, pp. 11--25 (1995)
12. Rubin H.A.: The Top 10 Mistakes in IT Measurement, IT Metrics Strategies, Vol. II, No. 11, (1996), <http://www.cutter.com/benchmark/1996toc.html>
13. Lavazza, L. and Barresi, G.: Automated support for process-aware definition and execution of measurement plans. In: 27th international Conference on Software Engineering (St. Louis, MO, USA, May 15 - 21, 2005). ICSE '05. ACM, pp. 234--243 New York (2005)
14. Abdullah, H. M. and Bhatti, A. M.: Deciding on Optimum set of Measures in Software Organizations". Master's Thesis MSE-200901, Blekinge Institute of Technology, Ronneby, Sweden. pp. 1--66 (2009)
15. Principles of heuristics optimization, <http://www.mm.helsinki.fi/kurssi/Marv/MSUU14/Heuristic.pdf>.
16. Rina, D.: Constraint Processing. Morgan Kaufmann (2003)
17. Fenton, N. E. and Fleeger, S. L.: Software Metrics: A Rigorous and Practical Approach 2<sup>nd</sup> Edition, Course Technology (1998)
18. ISO/IEC 9126-1 International Standard 1<sup>st</sup> Edition 2001: Software engineering — Product quality — Part 1: Quality model. Reference Number ISO/IEC 9126-1:2001(E) (2001)
19. ISO/IEC 9126-2 International Standard 1<sup>st</sup> Edition 2003: Software engineering — Product quality — Part 1: Quality model. Reference Number ISO/IEC 9126-2:2003(E) (2003)
20. ISO/IEC 9126-3 International Standard 1<sup>st</sup> Edition 2003: Software engineering — Product quality — Part 1: Quality model. Reference Number ISO/IEC 9126-3:2003(E) (2003)

21. ISO/IEC 9126-4 International Standard 1<sup>st</sup> Edition 2004: Software engineering — Product quality — Part 1: Quality model. Reference Number ISO/IEC 9126-4:2004 (E) (2004)
22. ISO/IEC 25000 International Standard 1<sup>st</sup> Edition 2005: Software engineering — Software product Quality Requirements and Evaluation (SQuaRE) — Guide to SQuaRE. Reference Number ISO/IEC 25000:2005(E) (2005)
23. CMMI Product Team: CMMI for Development Version 1.2 (CMMI-Dev, V1.2): Improving processes for better products. CMU/SEI-2006-TR-008, ESC-TR-2006-008, Software Engineering Institute (SEI) (2000)
24. Lamouchi, O., Cherif, A. R., and Lévy, N.: A framework based measurements for evaluating an IS quality. In: Proceedings of the Fifth on Asia-Pacific Conference on Conceptual Modelling - Volume 79 (Wollongong, NSW, Australia, January 01 - 01, 2008). A. Hinze and M. Kirchberg, Eds. Conferences in Research and Practice in Information Technology Series, vol. 325. Australian Computer Society, Darlinghurst, Australia, pp. 39--47 (2008)
25. Thakkar, D., Hassan, A. E., Hamann, G., and Flora, P.: A framework for measurement based performance modeling. In: Proceedings of the 7th international Workshop on Software and Performance (Princeton, NJ, USA, June 23 - 26, 2008). WOSP '08. ACM, pp. 55—66 (2008)
26. Kaner, C. and Bond, W. P.: Software Engineering Metrics: What Do They Measure and How Do We Know?. In: *10<sup>th</sup> INTERNATIONAL SOFTWARE METRICS SYMPOSIUM, METRICS*, (2004) <http://www.kaner.com/pdfs/metrics2004.pdf>.
27. Murty, K. G.: Optimization Models For Decision Making: Volume 1. Dept. of Industrial & Operations Engineering, University of Michigan, Ann Arbor, USA. (2003) [http://www.ioe.engin.umich.edu/people/fac/books/murty/opti\\_model/junior-0.pdf](http://www.ioe.engin.umich.edu/people/fac/books/murty/opti_model/junior-0.pdf).