THE BENEFITS TO MANAGEMENT OF USING GQM, CONTINUOUS GQM, AND V-GQM IN A MEASUREMENT PROGRAM

A thesis submitted
To Kent State University in partial
Fulfillment of the requirements for the
Degree of Master’s of Science in Computer Science

By Roger Kurian
August 2009
# TABLE OF CONTENTS

ACKNOWLEDGEMENTS (iii)

1 INTRODUCTION (1)

2 THE GOAL QUESTION METRICS (GQM) STUDY (3)
   2.1 What is GQM (3)
   2.2 Significance of the GQM (4)
   2.3 GQM Implementation (5)
   2.4 GQM Evolution (5)

3 VARIATIONS TO THE GQM STUDY (6)
   3.1 Continuous GQM and V-GQM (6)

4 DATA WAREHOUSING (8)
   4.1 What is a data warehouse (8)
   4.2 Significance of a data warehouse (9)
   4.3 Practical implementation of a data warehouse (9)
   4.4 Construction of a data warehouse (10)
   4.5 Data collection for a data warehouse (10)

5 EVALUATING A REAL-LIFE PROJECT (11)
   5.1 The GQM Approach (12)
   5.2 The GQM method phases (12)
      5.2.1 The Planning Phase (13)
      5.2.2 GQM Definition Phase (16)
      5.2.3 Data Collection Phase (21)
      5.2.4 Interpretation Phase (27)
   5.3 Application of continuous and V-GQM to the real-life project (28)
   5.4 Application of GQM/variations to find out the rate/amount of data (31)

6 MANAGEMENT BENEFITS BY USING GQM/VARIATIONS (32)

7 CONCLUSIONS (33)
   7.1 Overall Research (33)
   7.2 Future Work (34)

8 BIBLIOGRAPHY (35)
ACKNOWLEDGEMENTS

I sincerely thank my advisor Dr. Austin Melton for his advice, guidance and continued support. Dr Melton has been instrumental in keeping me focused in my research.

This thesis is dedicated to the memory of my late father: Mr. N.S. Kurian.

I also thank my family for their love and support without which I would not have reached this stage in my career.

Finally I thank Dr Arden Ruttan and Dr. Paul A. Farrell for agreeing to be on my thesis committee despite their busy schedules.
1 INTRODUCTION

In today’s world, software measurement programs are very important [27]. There are many issues to consider when a company wants to start a measurement program. These issues include cost, time, risk, management buy-in, user buy-in, and effectiveness of the program. Unfortunately, there is a dichotomy between the quantity of developed metrics and those used [25]. A common challenge faced by many projects, programs, and/or organizations is the initial development of metrics. In order to initiate metric development a methodology must first be selected that will yield a set of logically accepted measures that are consistent with the project's business philosophy. GQM addresses the above issues. GQM tries to make sure that the metrics used, and thus, the measurement program, are effective [13]. Instead of choosing metrics because of what other companies have done or what metrics are popular, when using GQM, metrics are chosen because they answer questions which when answered can help the company achieve its goals. The GQM methodology provides a great deal of insight to management in terms of choosing the right kind of metrics to be used for the measurement program. Thus, the measurement program should be effective (assuming there are metrics which do, in fact, answer the questions, and the questions have been chosen well).

However, even if the metrics are effective in answering important questions, a metrics program may not be successful if users (i.e., persons who implement the measurement program) and especially, if managers do not like the measurement program. Reasons why users and/or managers may not like measurement programs include because the programs may be hard and difficult to implement, because they are too much work, and because the users and especially, the managers do not understand the connections between the metrics and what the company is doing. Software metrics don’t solve problems; people solve problems [14]. However, GQM can be used to show the connections between what the company is doing and the metrics, and using continuous GQM and V-GQM compared to just GQM is much easier.

Continuous GQM and V-GQM have mainly been designed as a way to improve the use of metrics. In fact, the purpose of GQM was to make metrics programs more effective in terms of making wise choices of the metrics used. Too often when people started planning to use metrics, they often did not put much good thought into which metrics to use. The purpose of GQM was, thus, to make people think about what they wanted to or needed to accomplish and then choose metrics that would further those goals. Thus, GQM, continuous GQM and V-GQM were developed to help in the choice and use of good metrics. However, continuous GQM and V-GQM can be used to make a metrics program easy to set up from managers’ and users' points of view. The focus then need not be on choosing the best metrics (though that should happen using GQM); the focus could be on how GQM and especially V-
GQM can help a metrics program function more smoothly from managers’ and from users’ points of view.

GQM shows management how metrics are useful because they are tied to goals. The GQM methodology is described in great detail in Chapter 2. Extensions to the GQM methodology such as continuous GQM and V-GQM are less time consuming and easier to understand because, for example, with V-GQM the metrics are put in place and left there until the goals are reached. Details around continuous GQM and V-GQM methodologies are explained in detail in Chapter 3. By the help of an example project in Chapter 5, I will show how easy and natural using the above methodologies can be especially V-GQM. The example project chosen for the purposes of this thesis involves the transfer of data from a source data warehouse to a destination data warehouse and front-ending the destination data warehouse with an online front-end web-based application using ASP.NET. Since this project involves a data warehouse, a few details around data warehousing have been provided for the benefit of the readers in Chapter 4. Finally, we approach a GQM study with the goal to find out the amount/rate of data in a data warehouse. The output of this helps us analyze the infrastructure/servers etc. and improve the efficiency of the data warehouse. This is explained in further details in Chapter 5. Chapter 6 summarizes the benefits of GQM, continuous GQM, and V-GQM for management based on my project.

Thus, by the use of a real-life project, each of the above methodologies has been put to use and benefits derived by management by the use of such methodologies has been documented. Other potential benefits to management by the use of such methodologies have also been highlighted as a part of this research. Management/organizations can thus take advantage of these methodologies and apply them to several measurement programs.
2 THE GOAL QUESTION METRICS (GQM) STUDY

2.1 WHAT IS GQM?

Goal Question Metric (GQM) is a software engineering methodology developed by Basili et al in 1994 [21]. This methodology works on the concept of defining goals, devising questions to find out the extent to which these goals have been met and capturing the answers to these questions by using metrics [1]. The information captured in the metrics by using GQM is tied into the management goals. Thus, management has a better understanding of what this information is and can put it to use to evaluate these goals.

An important point to be noted here is that the GQM method is used to evaluate a particular project; the GQM method itself does NOT improve the project. However, the results of GQM evaluation can indeed be used by the architects/developers towards the evaluation of the project. Thus, the goal of a certain project could be to improve a particular process within that project, and the GQM method will allow us to evaluate the extent to which that process improvement has occurred. However, it is for the project designer, architects and developers to improve that process. It is critical to understand this distinction in the use of GQM methods in order to understand this method in finer detail.

The GQM methodology involves goals and objects:

Goals: These are often related to user requirements,

Objects: Each and every part of the software architecture.

Basili et al in 1994 defined the GQM process as follows[21]: GQM presents a systematic approach for integrating goals to objects: models of the software processes, products and quality perspectives of interest are based upon the specific needs of the project and the organization. “Objects” are the areas that need to be considered in order to meet the defined goals. As an example, let’s say a goal is to organize data in a server such as to increase server capacity by 20%. Then “Objects” would be existing data in the server that could possibly be compressed, current capacity of the server that could possibly use partitioning in order to improve capacity, etc. Thus, “Objects” are the KPI (Key performance indicators) that need to be considered in order to attain the Goals.
Thus, when there is a need to evaluate a certain process within an organization, we first need to define the goals. During the application of the GQM method, these goals will be refined into questions whose answers will let us know if the goals have been accomplished. The metrics are used in that the measurements we get from the metrics will supply all the necessary information for answering the questions.

Thus, the GQM approach provides a framework involving three steps:
- The major goals of the development project
- Questions derived from goals that must be answered in order to determine if the goals are achieved.
- Measurements that provide the most appropriate information for answering the questions.

This thesis focuses on the application of the GQM methodology to a real life cycle project (STC/SEI Interface to Odyssey) in its several phases: Planning Phase, Definition Phase, Data Collection Phase, and Interpretation Phase.

### 2.2 SIGNIFICANCE OF USING GQM

The approach was developed by Dr. Victor Basili and colleagues in conjunction with their work at the NASA Software Engineering Laboratory (SEL), was refined during the 1990s, and now, serves as the foundation framework for many measurement initiatives [3],[22]. Though designed for use with software development, GQM can be applied to all life-cycle products, processes, and resources, and is well aligned with an organizational environment. It is an appropriate means to achieve reliable empirical data and knowledge about an organization’s development and organizational practices to drive systematic process improvement.

A primary focus of using GQM as developed by Basili and his colleagues is the definition of goal-driven metrics. GQM can also address data collection, analysis and interpretation and packaging experiences for use in future initiatives. These activities are as important as defining the metrics because they guide how the data are actually used.

The GQM approach has been used with software life cycle projects in the past but was not specifically devised or designed to work with software projects using data warehouses (as seen in the example project). GQM has been used with data warehouses for quality management of the data warehouse and in combination with a tool called QFD (Quality Function Deployment) in order to evaluate a data warehouse. However, the key idea of describing GQM (and its extensions) as a part of this thesis is to show how management can derive benefits by the use of this technology in order to better understand metrics used in the measurement program of such projects.


2.3 GQM IMPLEMENTATION

The concept of GQM has evolved tremendously over the last few years with its emphasis on the improvement of several organizational processes. The GQM Method for any organizational process, according to Basili, can be implemented using the following six steps [24]:

1. Develop a set of goals
2. Generate questions that define those goals as completely as possible in a quantifiable way
3. Specify the metrics needed to answer the questions
4. Develop mechanisms for data collection
5. Collect, validate and analyze the data to provide feedback to projects for corrective action
6. Analyze the data in a postmortem fashion to assess conformance to the goals and to make recommendations for future improvements

The first three steps are about establishing a goal-driven measurement program where the identification of goals triggers the identification of appropriate metrics. The remaining steps focus on collecting and using the measurement results for better decision making.

2.4 GQM EVOLUTION

Studies [4], [5], [12] have shown that the GQM can be extended to be used in continuous iterative loops (continuous or iterative GQM) as well as used until the goals have been met (V-GQM). Thus, continuous GQM and V-GQM evolved out of the parent GQM methodology. However, they differ slightly from GQM, and these differences allow for different evaluations of a data warehouse when applied to data warehousing.

Both V-GQM and continuous GQM technologies have been applied sparingly to software life cycle projects. This research proposes the application of these methodologies to software life cycle projects and demonstrates potential benefits derived by management in using these advanced methodologies.
3 VARIATIONS TO THE GQM STUDY

3.1 CONTINUOUS GQM AND V-GQM

As described earlier, the GQM methodology works on the concept of defining goals, devising questions to find out the extent to which these goals have been met and capturing the answers to these questions by using metrics. Studies [4] have shown that the GQM can also be used iteratively so as to introduce a feedback loop and thereby, for example, evaluating a data warehouse at predefined intervals. This is the continuous GQM approach and is of much significance to this research. Continuous GQM is an extension of the GQM methodology that has sparingly been used with software projects.

The continuous GQM methodology applies GQM iteratively over specific intervals of time. This helps us realize the extent to which the goals have been met at certain specific intervals in time over a given timeframe. For example, let’s says the year-end goal of an organization is to improve the quality by 50%, and this goal cannot be realized in a single quarter (Q1). It may take the organization all 4 quarters of the year (Q1, Q2, Q3 and Q4) to realize the above goal. In such a scenario, the continuous GQM methodology works in an iterative loop fashion as shown in study [4] to evaluate the quality over the four quarters Q1, Q2, Q3 and Q4, and tells the architects/designers to what extent the goal has been realized in each of the above quarters i.e. quality could have increased to 20% in Q1, 30% in Q2, 40% in Q3 until the Goal is finally met i.e. 50% in Q4. In the above scenario, if instead of the continuous GQM only the GQM methodology were applied, then it would need to be applied at the end of the year which would provide little or no information to the management about the intermediate results towards goal accomplishment. If GQM is applied to every quarter, at the end of every quarter, the GQM team would need to reassess the goal, to check to see if it needs to be applied again. Thus, the continuous GQM used merely one time would be able to provide all information to the management regarding the Goals. The very word continuous implies that for its effect to be realized, this has to be applied more than one time. The difference between simple GQM and continuous GQM is that every time a simple GQM study is done, then we need to check to see if the goals have been met or not or the extent to which the goal has been met and then reapply GQM versus in continuous GQM, we do not need to continually assess the extent of goal achievement as it progressively continues for the programmed number of times that we have set it to function for. This indeed saves a lot of time and effort to the management and helps management better assess the metrics derived as an output to these iterations. The continuous GQM methodology, by evaluating the goals in an iterative loop fashion over specific
intervals of time (4 Quarters in this example), provides the extent of goal improvement and tells the developers/architects the level of effort that needs to be put in so that the defined goals could be realized at the end of the year.

Thus, the continuous GQM method allows for the rate of improvement of certain specific goals to be analyzed and provides for more comprehendible metrics in a measurement program.

An important point to be noted here is that the continuous GQM will only work for the specified number of times that it has been programmed to work and will not continually evaluate the process until the goals have been achieved (based on studies done)[4].

However, an interesting proposal to this thesis (based on ideas derived from studies done) [12] would be to use this methodology progressively until the final goal of the organization has been met. This method could use a predefined counter for the goal (value of which is fed in by GQM team) and continually validate the extent of goal improvement to the value in that counter. When the output of the GQM iteration either meets or exceeds the value in the counter, this method would stop applying any further iteration. Thus this method continues to apply the GQM methodology progressively until the goal has been met or exceeded. This provides far more benefits and ease to the management than merely applying the GQM or continuous GQM methodology as it merely needs to be applied once and it will continue until the goal has been met. This is an interesting idea which extends the concepts of continuous GQM and since this methodology validates the goal at the end of every iteration we call it validating GQM or V-GQM.

By application of the above methodologies, there is a good possibility that the goal might have exceeded. In such a scenario, the validating GQM or V-GQM methodology can again be used (but this time for goal refinement) in a way described in reference [5]. The Validating GQM (V-GQM) methodology for goal refinement works on the idea of introducing a GQM feedback loop for refining the goals set by the organization. The V-GQM study works in a bottom up fashion and introduces a life cycle perspective, creating a process, spanning several GQM studies to refine the goals set by the organization. Thus, when the goal is exceeded, the new exceeded value of the goal can be used towards the subsequent GQM study.
4 DATA WAREHOUSING

4.1 WHAT IS A DATA WAREHOUSE?

Note: To apply the above concepts of GQM, an example project using a data warehouse has been chosen. Therefore, for the benefit of the readers, the subsequent sections describing data warehouses have been added to this thesis.

DEFINITION

The term data warehouse was coined by Bill Inmon in 1990. According to Inmon a data warehouse is a "subject-oriented, integrated, time-variant, nonvolatile collection of data in support of decision making"[15].

In simple words a data warehouse is a collection of databases that gathers data from the various operational systems and is typically loaded from these systems at regular intervals.

The data in the data warehouse is organized for use by analytical applications and user queries. Data warehousing has been widely used in the field of information technology so that organizations can effectively use digital information for business planning and decision making.

A data warehouse is different from a data mart which is a database, or collection of databases, designed to help managers make strategic decisions about their business. A data warehouse is a collection of databases across an entire enterprise, whereas data marts are usually smaller and focus on a particular subject or department. Some data marts, called dependent data marts, are subsets of larger data warehouses.

Data warehouses contain a wide variety of data that present a coherent picture of the conditions in an organization at a single point in time.
4.2 SIGNIFICANCE OF A DATA WAREHOUSE

Data warehousing has been widely used in the field of information technology so that organizations can effectively use digital information for business planning and decision making [16]. In the realm of information management, everyone at some point comes across the data warehouse phenomenon. It is therefore important to understand the data warehouse architecture since it plays or will play a vital role in understanding the roles and responsibilities of a data warehouse architect and the whole information management scenario.

4.3 PRACTICAL IMPLEMENTATION OF A DATA WAREHOUSE

A data warehouse is generally set up on an enterprise mainframe server. The data in the data warehouse database is organized for use by analytical applications and user queries. The data warehouse database contains data from various online transaction processing (OLTP) applications and other sources and is selectively extracted and placed in the database. An online transaction processing (OLTP) application is a program that assists in the processing of a transaction and provides all the necessary information to complete the transaction [17].

Therefore, the concept of data warehousing emphasizes the capture of data from diverse sources for useful analysis and access, but does not generally start from the point-of-view of the end user or knowledge worker who may need access to specialized, sometimes local databases.
4.4 CONSTRUCTION OF THE DATA WAREHOUSE

In general, a data warehouse is planned in the very same way as one would plan to design any computer application [18]. Step one of the planning process begins with querying the users to determine the scope of the data warehouse and what high level business requirements need to be accomplished. After defining the scope, a data warehouse team of business users and information professionals compiles a list of different types of data that should go into the warehouse. Once the business requirements have been gathered and validated, data elements are organized into a conceptual data model. The conceptual model is used as a blueprint to develop a physical database design. Finally, as in all systems design projects, there are a number of iterations, prototypes, and technical decisions that need to be made between the steps of systems analysis, design, development, implementation, and support. There exist two generic design techniques for constructing the data warehouse: the top-down approach and the bottom-up approach.

4.5 DATA COLLECTION FOR THE DATA WAREHOUSE

In order to collect data for the data warehouse, the data warehouse team must determine the kind of data that needs to be warehoused. The next step would be to find out where such data can be found. Some of the data will be internal to an organization. In other cases, it can be obtained from other sources which may be external to the organization. Another team of analysts and programmers creates extraction programs to collect data from the various databases, files, and legacy systems. At this point, they try to ensure that the data has no errors (cleansing), and then copy it all into the data warehouse. This source data extraction, selection, and transformation process is unique to data warehousing. Source data analysis and the efficient and accurate movement of source data into the warehouse environment are critical to the success of a data warehouse project [19].
5 EVALUATING A REAL-LIFE PROJECT

It is critical to understand the GQM methodology and its use with software life cycle projects first before analyzing the variations of the GQM methodology. In this section we focus on the application of the GQM methodology to a real life cycle project (STC/SEI Interface to Odyssey) in its several phases. The idea behind this project is twofold:

1) To bring data from the STC data warehouse onto the Stanford data warehouse also referred to as Odyssey.
2) The Odyssey data warehouse would then need to be front-ended by an online ASP.NET web-based application called SWP (Stanford Wealth Platform) Portal.

This was a project that I worked on for Odyssey from Jan 2008 through Oct 2008 in my capacity as the lead database architect/project manager with a client company called Stanford Financial. During the course of this project, in order to understand the GQM methodology further, I took the liberty to use it to this real life cycle project. This helped me evaluate to what extent the data transfer between the two data warehouses occurred at specific points in time as well as the extent to which certain goals (Time, Risk, Cost) set by the organization were met. The outputs of these evaluations were sent in the form of status reports to senior management. As lead database architect on this project, I was involved in the ETL (Extract, Transformation and Load) process (in addition to other roles/responsibilities) for this project. The data had to be first extracted from the STC-SEI data warehouse. The extracted data was in the form of pure XML files which then needed to be transformed into the SWP (Stanford Wealth Platform) format which was pipe delimited .txt files and then loaded onto the same. Further work involved front ending the Odyssey data warehouse with the SWP portal and populating the SWP fields with data from the Odyssey data warehouse. The GQM methodology applied to this project was done purely for the purpose of this thesis and to help me better understand this study. Some of this work was used by the firm to find out the extent of data transfer across the two data warehouses and the extent to which certain goals (Time, Risk and Cost) of the organization were met.
5.1 The GQM Approach

GQM defines a measurement model on three levels [25]:
- **Conceptual level (goal):** At this level, a goal is defined based on the various aspects of quality that needs to be accomplished for a variety of reasons from various perspectives and relative to a particular environment.
- **Operational level (question):** A set of questions are developed in order to characterize the assessment or achievement of specific goals.
- **Quantitative level (metric):** A set of metrics associated with every question is found or designed in order to answer it in a quantitative way.

5.2 The GQM Method Phases

**Legend for font:** **Theory:** Regular font, **Application in example project:** Italic font

The GQM method contains four phases[25]:

1. The **Planning phase**, in which we create a project plan by selecting the project for measurement, defining it, characterizing it and finally planning the entire project.
2. The **Definition phase**, in which the goals, questions, metrics and hypotheses are defined and documented.
3. The **Data collection phase**, during which the actual data collection takes place, resulting in collected data.
4. The **Interpretation phase**, during which the collected data is processed with respect to the defined metrics into measurement results that provide answers, to the defined questions, after which goal attainment can be evaluated.
5.2.1 The Planning Phase

The GQM planning phase basically deals with the logistics of implementing GQM and key plans that need to be documented. We begin the planning phase by collecting all the necessary information and motivating the “GQM Team” for a successful start of this program. The aim of this phase is to construct a project plan which basically contains documentation of procedures, schedules and objectives of a measurement program. The planning phase is performed to fulfill all the necessary requirements to make the GQM measurement program a success; including training, management involvement and project planning. The planning phase can be divided into four sub-phases:

1. **GQM Team**

   There are a few objectives that need to be taken into account when establishing a GQM team: The GQM team should be knowledgeable on the subject. It should be objective and improvement oriented since it is in charge of planning the whole project. The GQM team should take care of data collection and preparing everything that is necessary for data interpretation. *Example GQM Team: Business Users, Project Manager, Business Analyst*

   *(Note: In most cases, the GQM team is a subset of the project team but does not include the entire project team)*

2. **Selecting Improvement Areas**

   The selection of the improvement areas basically targets the goal of the GQM process. If the goal is to improve a certain software process or a product, selection of improvement area should be performed with respect to that goal taking into account the cost, time, risk and quality. The GQM team should consider all the details, like problems that might occur, all external influences, people, processes and products involved, and the previous knowledge about measurement by the people who are going to participate in this project. *For the example project chosen, the improvement areas included: Time, Risk and Cost.*

3. **Selecting Application Project and Establishing Project Team**

   There exist several people who work on the development of a certain software project. All of these people are part of a software project team. Since the success of a measurement program heavily depends on the project team members, the GQM team
should make an effort to align measurement objectives and improvement ideas of the project team. It is imperative that everyone in the project team understands the theoretical background of the GQM process. However, there is also a need to emphasize the practical aspect of GQM in relation to the project. That also means that they should clearly know the details of the goals and know other possible benefits of the measurements. Application Project: STC/SEI Interface to Odyssey (data warehouse from STC to be brought over to Odyssey data warehouse) and front-ended by the SWP Portal. Project Team: Business Users, Project Manager, Business Analyst, Developers/Data Warehouse Designers and Architects, Testers.

4. Project Plan

As the final step to this phase, the GQM team creates the project plan which should contain the following pieces of information:

Measurement program: The idea behind this project is to bring over the accounts/data that are custody in the STC (Stanford Trust Company) data-warehouse into the Odyssey data-warehouse which needs to be front-ended by the SWP Portal.

Introduction (It presents a brief overview and background of the measurement program. It basically contains explanation to improvement ideas that are related to software development project goals): Stanford has several custodians. Custodians are the different Stanford Locations that have custody of Stanford Funds. Customers come to these locations to deposit their funds. Stanford has custodians in several different parts of the globe. These include SIBL (Stanford International Bank Limited), STC (Stanford Trust Company) and SCB (Stanford Coins and Bullions). The example below describes the process of bringing all accounts from one of Stanford’s data warehouses (STC) onto Odyssey/SWP Portal.
Current Software Process Model:

As shown in the diagram, STC is currently utilizing a reporting platform, Trust3000, managed by SEI. SEI is a leading global provider of outsourced asset management, investment processing and investment operations solutions.

Schedule: The schedule gives a complete description of tasks that should be performed, types of resources to be used, the time period where in they can be used, results and expected costs and benefits. Details around the schedule are mentioned on page 26.

Organization: This describes relevant organizational objectives in the measurement program. In the project we are studying, there are several organizational objectives in this program. However, the key organizational objectives on which we are going to focus are to time, risk and cost.

Management process: (Contains priorities, and descriptions of reporting procedures and risk control activities): Once the interface goes live, reporting will happen through the SWP Portal. Risk control activities will be put into place during the course of interface development.

Training and Promotion: The Stanford Financial Advisors need to be trained to use the Stanford Wealth Management Platform: Odyssey
5.2.2 GQM Definition phase

The main task of the definition phase is to provide formal definition of measurement, which includes definitions of questions and hypotheses, reviewing, checking and producing GQM, measurement and analysis plans. The GQM definition phase can be divided into 4 sub-phases [25]:

1. **Goals Definition**

In this sub-phase, goals are formally defined and structured on the basis of improvement. The following questions were asked while defining the measurement goals, namely: What is the Project for which we need to define goals? A: The STC/SEI interface to Odyssey (SWP). Why do we need to understand, control or improve the Project? A: In order to meet the goals of time, risk and cost. What aspects of the quality focus of the Project does the goal focus on? A: Schedule, percent of Errors, Final Product Functionality, volatility of the requirements, budgeted costs. Who are the people involved in the assessment of the goals? A: PM/Business Owners/ Testers/ Business Analys.. What is the context or the environment in which the goal assessment takes place? A: UAT Environment.

In selection of relevant goals some objectives that need to be taken into account are: Selecting the most important goal: Time. Considering factors that influence these goals: Resource Constraints. Methods of improving the goals: Simultaneous/Planned Resource Management, Selection of suitable goals of highest priority: Time, Risk and Cost

2. **Software Process Model**

Either a revision of the existing software process model or a new one, complete and consistent with the definitions of the project, could be used. The earlier version of the software model portrays the system that currently existed. Below is the proposed version of the software model.
Proposed Software Process Model:

Note that post implementation, the night cycle feed from SIBL that currently feeds STC will feed Odyssey. Reporting will not occur through SEI but instead through the SWP portal that will front end the Odyssey system.

3. **GQM Interviews**

Since the communication between the GQM team and the project member team/development team is essential for the project's success, the capability of the GQM team to extract knowledge from the project team experts is extremely important. (The development of the whole measurement program is based on this knowledge). One common way of communication between the GQM team and project team members is
individual interviews. An interview can be very useful since it can contain all the main issues that one should focus on, such as: How knowledgeable are the project members in the team? 

*A: The project team members have all knowledge about the project details and functionality.*

Which external factors are expected to influence the metrics and how are they going to affect the transfer of knowledge/understanding between the business team and the development team? 

*A: The project/development team is focused on completing the project on time and making sure all requirements are met. The business team, in addition to the above is also focused on the making sure the other goals of risk and quality are met.*

The GQM interviews help us understand the mindset of the various teams in the project as well as helping make the different teams aware of the overall goals which are assessed on the completion of the project.

4. **Question and Hypotheses**

The main idea in this phase is gaining operational definitions, i.e., a question is a goal refined to an operational level. It must be emphasized that the question should not be too detailed or too abstract, but intermediate which will provide an optimal interpretation. Example of a general question could be: What is the extent of change in the requirements of this project? A detailed question on the other hand could be which specific requirements in this project changed since its initiation? A right question is the one focused on an intermediate level that does not require us to scan through the entire project. One such question could be what is the amount of time required for development work/coding? Hypotheses are the expected answers which are going to be examined at a later stage in the GQM. It is important to ensure that the questions asked in this step tie into/relate to each of the above defined goals. First, it is important to define questions otherwise they will not represent goals. Moreover we are interpreting results in the terms of hypotheses and therefore, many misinterpretations are likely to occur. Hence, it is necessary that the questions and hypotheses should be reviewed and, if necessary, reformulated.
Questions and Hypotheses created for this project:

MSS Metrics Base (Containing all questions and answers):

Parent Goal: Time (Based on the schedule and milestones (amount of effort) defined for different phases of the project)

Goal (G1): To predict the schedule in order to manage it

Question (Q1): What is current schedule for the project?

A: Based on the dates provided between Jan 2008 through Oct 2008, the bar graph shown in the section MSS Analysis Sheet: Current schedule for the project is plotted for the different phases of the project.

Goal (G1.1): effort

Question (Q2): Does the amount of effort to attain the milestones follow the path of the predicted effort by management?

A: Amount of effort in hours is plotted against the predicted effort by management as shown in the section MSS Analysis Sheets: Expected vs Actual Effort.

Parent Goal: Risk (This goal is dependent upon the stabilization of requirements, the amount of time taken to locate and remove errors that are due at UAT and on certain specific modules at high risk)

Note: UAT stands for user acceptance testing and it is a series of testing done by the users of the software using test scripts (provided by the testers) to check to see if every module of the software is completely functional under the various real world scenarios that the software can potentially be used for.

Goal (G2): Risks associated with the volatility of the requirements

Question (Q3): What is the volatility of the requirements? At what point do the requirements stabilize?
A: I have plotted the graph (shown in section MSS Analysis Sheets: Requirement Volatility) based on the total number of requirements (on my project) per month for the first 7 months: Jan 2008 through July 2008. The requirements stabilized on the 6th month and from that point on, there were no scope creeps (requirement changes). This was good for the project and we did not need to go through any change management cycles to undertake any new requirements post June 2008.

Goal (G 2.1): Errors involved in the project

Question (Q4): When would most errors be located and removed? All errors should be located and removed prior to User Acceptance Testing: UAT.

A: I have plotted the graph shown in the section MSS Analysis Sheets: Error location and removal based on the total number of errors encountered per week for the first 12 weeks: Feb 2008 (Note that Jan 2008 was spent in requirement gathering and development effort only started in Feb 2008) through April 2008. This project was scheduled to enter in UAT (User Acceptance Testing) in May 2008 and at the end of the first 12 weeks (April 2008) 142 errors were found. This project was expected to enter into UAT at week 12 and 142 errors were found and removed.

Parent Goal: Cost (Based on budgeted values for the 10 modules of the project)

Goal (G3): Cost

Question(Q5): Do any of the modules exceed the budgeted cost? If so, how many?

A: It was found, that out of the 10 modules required to be developed for the SWP portal, 3 modules exceeded budget. As a consequence, these 3 modules were developed as a part of the subsequent release in Jan 2009.
5.2.3 Data Collection Phase

Once the definition activities are complete the actual measurement can start. The success of every project depends upon acquiring accurate measurements. Sometimes the product measurements can be obtained without human intervention- automatic and digitized analysis. But in the case of process and resource measurements, this is usually not possible. Manual data collection could cause some problems due to high demands and rigorous procedures. However, this is usually the most common and the easiest way of collecting data although it sometimes takes a lot of effort to set in place the procedures for accurate measurements. For this project, a manual data measurement program would be the best alternative. The result of the data collection phase are filled in forms and stored safely.

Hold Trial Period

A trial measurement period should be held in order to avoid mistakes and to test the data collection procedures, tools and forms, before the actual data collection period. Next, a session should be organized in which all people participating in the measurement program should approve all the plans, forms, tools and procedures. The tool used to capture the questions and answers will be a Microsoft Excel worksheet. As discussed above, the members of the measurement program will be the entire GQM Team: Business Users, Project Manager and Business Analyst.

Metrics Base

Data collection forms should be filled in and then gathered by GQM team, which is responsible for checking correctness of the filled-in data. Should any messages or mistakes occur, they are supposed to be corrected immediately. MSS stands for Measurement Support System and is a tool used for collecting, storing and maintaining data. The MSS contains two tools: The MSS Metrics Base and the MSS Analysis Sheets. A metrics base is the first part of the measurement support system (MSS) and plays an important role in data collection. The MSS metrics base contains all data collected during the questions and hypothesis step and the answers to these questions that are obtained as a result of GQM interviews, questionnaires passed on the project team, and GQM team sessions with the project team. The MSS Analysis sheets are various kinds of graphs, charts and other presentation materials based on the collected data and differentiated by levels of abstraction. The levels of abstraction, listed in ascending order are the row data layer, processed data layer and graphs and charts layer. Each analysis sheet should have a description of a specific goal, all derived questions (as in the GQM plan) and all of the data sufficient for answering the questions in a satisfactory way, with respect to the goal.
During MSS development, spreadsheet tools (MS Excel) are used to store, maintain and process the collected data, and the presentation tools are used to prepare the collected data for presentations.

*Based on the above description of the Measurement Support System (MSS), I have put together a MSS worksheet with the two tabs mentioned below:*

- **MSS Metrics Base (Containing all questions and answers): Page 23**
- **MSS Analysis Sheets (Containing collected data and plotted in the form of graphs and charts)**

**MSS Analysis Sheets:**

**Question 1: Current Schedule for the project**

*Based on the dates provided by Stanford between Jan 2008 through Oct 2008, the following bar graph is plotted for the different phases of the project.*
**Question 2: Expected VS Actual Effort:** Effort is usually measured in hours worked on specific project tasks, such as training, requirements, design, coding, and testing. *Amount of effort in hours is plotted against the predicted effort by management. Projects that generally increase in effort steadily, ramp up to 100% effort midway and steadily drop are generally expected to meet milestones and complete on time. As can be seen, amount of actual effort in hours followed the path predicted by management.*
**Question 3: Requirement volatility**

Late requirement changes are costly and may cause a ripple effect and additional changes. The earlier in the Life cycle the requirements stabilize, the less the risk. Figure below shows total requirements per schedule, indicating stabilizing requirements. *I have plotted the below graph based on the total number of requirements (on my project) per month for the first 7 months: Jan 2008 through July 2008. The requirements stabilized on the 6th month and from that point on, there were no scope creeps (requirement changes). This was good for the project and we did not need to go through any change management cycles to undertake any new requirements post June 2008.*
Question 4: Error location and removal
This implies the ability to estimate the total number of errors in the software. One industry guideline is to expect approximately 7 errors per 1000 Source Lines of Code. This guideline is helpful in an overall estimate of the number of errors, but does not take into account the rate at which errors are removed.

I have plotted the below graph based on the total number of errors encountered per week for the first 12 weeks: Feb 2008 (Note that Jan 2008 was spent in requirement gathering and development effort only started in Feb 2008) through April 2008. This project was scheduled to enter in UAT (User Acceptance Testing) in May 2008 and at the end of the first 12 weeks (April 2008) 142 errors were found and removed.
Question 5: High risk modules

To answer question 5, inputs from multiple metrics are needed. One factor is the risk derived from the table below. The number of errors by criticality also serves as an input. The table below shows the different modules that need to be developed in order for the project (STC/SEI) to go live and the error criticality/risk associated with each of these modules (1 = low risk, 2 = medium risk, 3 = high risk). The column to the right (risk total) shows the overall risk of each of these modules.

<table>
<thead>
<tr>
<th>Module Number</th>
<th>Error Criticality low = 1</th>
<th>Error Criticality Medium = 2</th>
<th>Error Criticality High = 3</th>
<th>Risk Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td>2</td>
<td>Medium</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td>1</td>
<td>low</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td>3</td>
<td>high</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td>2</td>
<td>Medium</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td>3</td>
<td>high</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td>3</td>
<td>high</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td>3</td>
<td>high</td>
</tr>
</tbody>
</table>

Note that there were a total of 10 modules to be developed. 3 modules exceeded the budgeted cost for development and hence the above table shows the risk associated with only the other 7 modules.
4.2.4 The Interpretation Phase

GQM Interpretation phase is the final and essential phase in the GQM method. During the interpretation phase, the collected data are used for answering the stated questions and in knowing whether the goals are achieved. In other words, results of the measurements are discussed and conclusions are made in terms of measurement results. If hypotheses and conclusions are consistent, it implies that the goal has been achieved. Below are the tasks that need to be performed during this phase:

Prepare Feedback Session

The GQM plan contains all the necessary information for preparing feedback sessions. GQM members should prepare feedback material, such as analysis sheets, presentation slides, handouts and if necessary, some additional material: MSS Metrics Base, MSS Analysis Sheet in Section 5.2.3 Data Collection Phase on Page 23,26. Feedback material should be very useful to project team members during feedback sessions. During these feedback sessions, the project team members are supposed to analyze, interpret collected data, draw conclusions, and translate conclusions into particular actions. Project members should have a constructive and goal driven approach. They should focus on the following points: Data interpretation with respect to the questions, goals as defined in the GQM plan and drawing conclusions and action points. Taking into account the stipulated figures around each of the Goals described above, some of the Goals/Expectations were met and exceeded. However, due to the volatility of the requirements there were some Goals (Cost) that were NOT met. Subsequent sections show how these goals could be evaluated further to find out the extent to which they were met.

Results

After a feedback session, the GQM team writes a meeting report containing all relevant observations, conclusions and action points formulated during the session. This is generally a synopsis of the GQM Plan and conclusions regarding the goals that were achieved. In this project, all modules with medium or low risk were completed on time. 3 modules however exceeded costs due to the changes in requirements. This is explained in further detail in the following section: Cost and Benefit Analysis. If the goals were not achieved, we should make a list of action points to achieve these goals in the next iterations.
Cost and Benefits Analysis of a successful program

Attaining a goal no doubt is an essential element of success of a program. However, it is very important to evaluate whether the estimated benefits exceed costs. This is crucial from an economical point of view and therefore should be performed at the end of this phase. Based on the requirements/changes in requirements, errors/risks and modules exceeding the guidelines, we perform a cost benefit analysis to find out if the project benefits can be realized with the anticipated costs. Based on the changes in requirements, errors/risks the cost benefit analysis yielded a 70% return i.e. 70% of the modules (7 modules) were completed within the assigned cost however 30% of these modules (3 modules) exceeded the anticipated costs. The continuous GQM study in section 5.3 of this thesis shows how to evaluate the remaining 30% of these modules over specific intervals of time to find out the extent to which these modules yielded a successful cost benefit analysis.

5.3 Application of Continuous and V-GQM to the real-life project

Continuous GQM is the continuous application of the GQM methodology. There are two ways how this could be done:

1) As shown in Study [4], GQM is applied in a continuous and automated fashion for a programmed number of times which produces intermediate metrics towards goal achievement. These intermediate results help management better understand the metrics and the extent of goal achievement throughout the course of the project.

2) Another way of doing so, as proposed in this thesis (based on ideas derived from studies done) [12], is to continuously apply the GQM methodology as shown above, until the goals have been met or exceeded. This method utilizes a counter that stores the final value of the goal and continuously validates the output of each GQM iteration to the value in the counter. If this value is met or exceeded, this process stops. Since the output of each iteration is continually validated against the value in the counter, this method is called validating GQM or V-GQM.

As seen in the above project, due to the volatility of the requirements there were some goals that were NOT met (See section above: cost and benefits analysis). Among the 10 modules that needed to be developed, 3 modules exceeded the budgeted costs and hence could not be implemented as a part of the Oct 2008 release. Senior management made a decision to implement these 3 modules in a subsequent release scheduled for Jan 2009. I decided to use continuous GQM as shown in method 1) above for 3 iterations scheduled to occur at the end of each month for the months of Nov, Dec and Jan. Application of the continuous GQM for each of the 3 months produced intermediate results (at the end of each month) towards goal accomplishment. However, the cost required to complete all 3 modules again exceeded budget (which means the goal was NOT met). This release was therefore postponed to a subsequent release scheduled in Feb 2009.
The advantage of using continuous GQM (as shown in Method 1) however versus the regular GQM methodology is that it produces intermediate results towards goal accomplishment versus the regular GQM methodology producing results either at the end of the project or requiring several GQM applications (requiring an assessment of the results after every application) which typically involves a lot more effort than continuous GQM.

An important point to be noted here is that the continuous GQM (as shown in Method 1 above) will only work for the specified number of times that it has been programmed to work and will not continually evaluate the process until the goals have been achieved. This is why, this research proposes the use of validating GQM as described in Method 2) above to progressively apply GQM until the goals have been met or exceeded.

As mentioned above, all 3 modules scheduled to go live as a part of the Jan 2009 release exceeded budget and it was decided to implement these modules as a part of the Feb 2009 release. In order to evaluate the intermediate results, I decided to use validating GQM as shown in Method 2) above, progressively until the goals were met or exceeded. In order to do so, a counter was set up (in excel) that stored the final value of the goal (cost in our example), GQM was applied continuously at the end of each of the weeks during Feb 2009. The output of each iteration was compared to the cost value stored in the counter. It was found that at the end of the week 2/27 (prior to implementation) all 3 modules were fully functional and could be implemented in a cost value that was LESS than the value of the budgeted cost for these modules. This is an example of an exceeded goal. However, for the purposes of validating GQM as proposed in method 2), the final value of the goal was MET.

Validating GQM (V-GQM) can also be used in a scenario (like the one above) where the goal of an organization is exceeded.[5] The V-GQM study works in a bottom up fashion, as described below, to refine the goals set by the organization. V-GQM introduces a life cycle perspective, creating a process, and spanning several GQM studies. After the GQM study has been completed, an analysis step of the plan is initiated. The metrics are analyzed to investigate if they comply with the plan or have extended it, and also to investigate if the metrics collected answer more questions than those posed in the original plan. The questions derived from the metrics are then used to form the goals for the next GQM study, effectively introducing a feedback loop. By introducing the bottom-up approach, a structured analysis of the GQM study is possible when constructing several consecutive GQM studies.
THE V-GQM METHOD

In this method, firstly the goals (requirements) are stated. Second, the questions are defined (high-level design). Third, metrics are derived (low-level design). Lastly, the measurements are collected and the results analyzed (implementation). Normally, this is where a GQM or continuous GQM study finishes. At this point, in the V-GQM study, three more steps are introduced: Validating metrics, question analysis and goal refinement. In these three steps, the metrics are validated, the questions are analyzed and the goals are refined in a bottom-up fashion.


Thus the output of the example project above exceeded the goal of cost. In a scenario like this, V-GQM could also be used to re-evaluate the goal of cost and a similar forthcoming project in the firm could be implemented with a lower value of the cost (exceeded goal). Thus, this methodology helps management makes strategic decisions for re-evaluating their goals thereby saving time, effort and money.

This marks the end of the work done on the example project.
5.4 Application of GQM and its variations to find out the rate/amount of data in the data warehouse:

The GQM and its variations can also be used to find out the amount of data/rate of increase of data in the data warehouse.

One of the questions of the GQM study could be to find the amount of data in the DW. Another question could be to find the rate of increase of data in the DW over a period of time. GQM could be used to find the total amount of data in the DW and if the maximum capacity of data that can be accommodated in the DW has been met. Metrics derived/output to the above study can be useful to management in analyzing the infrastructure/servers, etc. If the maximum capacity of the servers has been met, it tells the architects that there is a need for additional server space and if the existing servers need to be replaced/appended by additional servers.

Variations of the GQM study (e.g. continuous GQM) can be applied to find out the rate of increase of data in the DW over a specific period of time thereby producing intermediate metrics towards the final goals.

This could help the DW designers/architects to more efficiently organize the data in the DW. Periodic checks on the servers and collectively/efficiently organizing the data in the DW also helps to run systematic queries over a more organized sample set of data. An organized sample set of data is created by making the appropriate table joins that link information across the different tables with the DW. This to a great extent improves the efficiency of the data warehouse.

Organizations can thus greatly benefit by using the GQM and its variations to find the rate/amount of data in the DW not just from an infrastructural perspective but also to improve the efficiency of the DW.

Thus, as proposed in this research, management in several organizations can use the above methodologies to their advantage not just for evaluating the goals until they have been attained but also for reassessing and refining the goals set by the organization using V-GQM.
6 BENEFITS TO MANAGEMENT BY USING GQM AND ITS VARIATIONS IN A MEASUREMENT PROGRAM

Although a direct benefit of GQM is establishing meaningful metrics, when GQM is applied within the context of systematic process improvement for software projects, it is particularly beneficial to management in the following ways:

- Understanding and base-lining the management organization’s practices
- Taking a structured and collective approach towards deriving metrics
- Guiding and monitoring processes
- Tracing collected metrics to pre-defined management goals
- Providing management in-depth insight and knowledge on the collected metrics.
- Assessing the overall results of the measurement program
- Helping management make strategic decisions towards the reassessment of goals
- Evaluating and certifying improvement activities

Benefits of the continuous GQM and V-GQM methodologies:

- Ease of use
- Saves time and effort
- Provides management intermediate results in a timely fashion
- Status reporting/milestone accomplishments
- Reduces technical and infrastructure related risks due to frequent evaluations
- Achievement of improvement goals
- Increased quality awareness and Quality Assurance (QA) involvement
- Increased capability to perform improvement initiatives
- Improved group synergy
- Financial gains
7 CONCLUSIONS

7.1 Overall Research

Organizations have several different kinds of measurement programs around cost, time, risk, quality, management buy-in, user buy-in, etc. In order to conduct a measurement program, information on the measurement objectives are captured in the form of metrics. In the past, there have been several problems reported with software metrics. The most critical of these is the fact that these metrics do not provide an easy way to be comprehended and applied. GQM is one such method that helps ask specific questions towards the management goals and provides for metrics that answer these specific questions. Since this metrics plan is defined purely based on management goals, GQM provides for better use of such software metrics by tying these metrics to the management goals. Thus, the GQM methodology provides for better understanding and base-lining the management organization’s practices by taking a structured and collective approach towards deriving metrics.

The GQM methodology provides for effective metrics that help answer important questions tied to management goals. However, even such an effective metrics program may sometimes not be successful. One of the most important causes is that such programs may be hard or difficult to implement. Variations to the GQM methodology such as continuous GQM and V-GQM are far easier to use and save the management time and effort.

The introduction of the GQM methodology to software measurement was a significant step forward in that, when used, it causes software measurement users to consciously consider the connections between what they want and need to accomplish and the metrics used. The continuous GQM variation was developed in conjunction with the automated use of metrics. The V-GQM variation used the idea of continuous GQM, though without requiring automated metrics, to highlight additional benefits of repeated uses of the GQM methodology. However, even with significant advances and improvements in software measurement programs, these programs will often not be successful without positive buy-in from managers and metrics users. Interestingly, managers and metrics users are not always most interested in the quality of the measurement programs. Often, especially, when busy, managers and metrics users really want a measurement program that in many ways is self sustaining and this self sustainability is a natural benefit of the continuous GQM and the V-GQM methodologies.
Of course, the most self sustaining metrics programs are in some ways the automated ones. However, especially, from managers’ perspectives, the continuous GQM and especially, the V-GQM are very self sustaining from a monitoring perspective. With the V-GQM methodology, once the metrics program is set up, it can run until the goal or goals are obtained. Such a program is ideal from a manager’s perspective. This self sustainability of using the continuous GQM and, especially, the V-GQM is an aspect of these methodologies which is a highlight of my using these methodologies in my data warehouse project. The measurement projects were easy for me to implement, and their use was transparent to my manager.

By the use of a real-life project, each of the above methodologies have been put to use and benefits derived by management by the use of such methodologies have been documented in this thesis. Other potential benefits to management by the use of such methodologies such as finding out the rate of increase of data and the amount of data in the data warehouse at any given point in time are also highlighted in this thesis.

Management/organizations can thus take advantage of these methodologies and apply them to several measurement programs.

7.2 Future Work

This research could be extended to do more investigating of data warehouse issues such as to find out the rate/amount of data in a data warehouse. The output of this could potentially help us analyze the infrastructure/servers, etc. and improve the efficiency of the data warehouse.

One could also do studies of metrics programs using and not using continuous GQM and V-GQM to see if the use of continuous GQM and V-GQM make things easier for management.

The result of such implementations could prove highly beneficial to the management for several different kinds of measurement programs.
8 BIBLIOGRAPHY


http://www63.pair.com/hefner/htdocs/metrics.doc

14) Linda Westfall, 2003: Are We Doing Well, Or Are We Doing Poorly?:  
http://www.westfallteam.com/Papers/Are_We_Doing_Well.pdf

Other GQM links:

Data Warehouse Definition:

15) DATA WAREHOUSES, DATA MARTS, AND DATA WAREHOUSING  
by Joseph M. Firestone, Ph.D:  
http://www.taborcommunications.com/dsstar/01/0320/102809.html  

16) Significance of a Data Warehouse:  
http://www.mnhs.org/preserve/records/dwintro.html

17) Practical Implementation of a Data Warehouse:


http://searchsqlserver.techtarget.com/sDefinition/0,,sid87_gci211904,00.html

18) Construction of the Data Warehouse:  
http://www.mnhs.org/preserve/records/dwintro.html

19) Data Collection of the Data Warehouse:  
http://www.mnhs.org/preserve/records/dwintro.html

20) Reliability of the Data Warehouse:  
http://www.mnhs.org/preserve/records/dwintro.html

21) What is GQM:  

22) Significance of using GQM:  

23) Benefits of the GQM:  

24) GQM Implementation:  

25) GQM Approach and method Phases:  
http://sel.gsfc.nasa.gov/website/exp-factory/gqm.html  

26) Software metrics and measurement principles:  
http://portal.acm.org/citation.cfm?id=181625

27) Software measurement program:  
http://www.totalmetrics.com/consulting-function-points/software-measurement-implementation