GQM+Strategies – Aligning Business Strategies with Software Measurement

Victor Basili[B,C], Jens Heidrich[A], Mikael Lindvall[B], Jürgen Münch[A],
Myrna Regardie[B], Adam Trendowicz[A]

[A] *Fraunhofer Institute for Experimental Software Engineering*
Fraunhofer-Platz 1
67663 Kaiserslautern, Germany
+49-631-6800-0

[B] *Fraunhofer Center for Experimental Software Engineering*
Maryland College Park
Maryland 20742
+01-301-403-0

[C] *University of Maryland*
Maryland College Park,
Maryland 20742, USA
+01-301-405-0

basili@fc-md.umd.edu, heidrich@iese.fraunhofer.de, mikli@fc-md.umd.edu,
muench@iese.fraunhofer.de, mregardie@fc-md.umd.edu, trend@iese.fraunhofer.de

Abstract

GQM+Strategies is a measurement approach that builds on the well-tested GQM approach to planning and implementing software measurement. Although GQM has proven itself useful in a variety of industrial settings, one recognized weakness is the difficulty for GQM users to link software measurement goals to higher-level goals of the organization in which the software is being developed. This linkage is important, as it helps to justify software measurement efforts and allows measurement data to contribute to higher-level decisions. GQM+Strategies provides mechanisms for explicitly linking software measurement goals, to higher-level goals for the software organization, and further to goals and strategies at the level of the entire business.

1. Introduction

GQM+Strategies is a measurement approach developed at the Fraunhofer CESE (Maryland, USA) and Fraunhofer IESE (Kaiserslautern, Germany). This approach is based on the familiar Goal Question Metric approach [1], [2], which is today in widespread use for creating and establishing measurement programs throughout the software industry. This new extension to GQM adds the capability to create measurement programs that ensure alignment between business goals and strategies, software-specific goals, and measurement goals.

Although GQM has served the software industry well for several decades, it never provided explicit support for integrating its software measurement model with elements of the larger organization, such as higher-level business goals, strategies, and assumptions. This lack of connection between the business goals and software is noticeable in several ways.

Software engineers are frequently faced with apparently unrealistic goals related to software development. For example, if the next version of a product with some embedded software needs to be released to the market in half of the originally planned time, the software development schedule is cut in half. There is rarely a discussion of trade-offs or other options to avoid catastrophic results. Software-specific business goals and strategies need to be defined explicitly and derived from business goals in a systematic and transparent way.

On the other hand, software development initiatives are not always linked to a clear contribution to business goals. Time and money are spent on software initiatives without them ever contributing to the bottom line of the business. There are no clear connections drawn between these software initiatives and the business goals and corresponding strategies. A measurement program that connects the two would make it more evident to software business management, facilitating more informed decisions.

0-7695-2886-4/07 $20.00 © 2007 IEEE
DOI 10.1109/ESEM.2007.66

Another aspect of the problem is that software improvement strategies, such as CMMI [3] and ITIL [4] are not directly linked to generating business value, thus it is not enough to merely follow such strategies, because they are strictly software strategies; the strategies need to be linked to business goals.

Thus, in practice, an approach to measurement that explicitly links high-level business goals and software measurement data is needed.

2. Current approaches

GQM+Strategies leverages the successful GQM approach, which is a pragmatic way of creating a measurement program based on software goals, the questions and models these goals generate, and the metrics that are needed to answer those questions. One benefit of GQM is that it assures that the metrics set is as small as possible and that the data collected are useful and address the defined software goals. However, GQM does not provide explicit support for dealing with goal dependencies and does not ensure that the totality of goals forms a rich set of relationships. This identification and use of the relationships between different measurement goals, as well as between measurement goals and higher-level business goals, is addressed in the extended GQM approach, GQM+Strategies.

There is a recent trend in the IT governance domain and the IT service domain that emphasizes the need for linking business goals and IT goals. However, the solutions proposed by models in these domains (especially COBIT® 4.1 [5] and ITIL release 3) only offer simple connections between predefined sets of goals and are rather focused on providing an IT infrastructure than linking software development or software engineering with business strategies.

We have found no other single approach to support the continuum of defining a full and useful range of goals. However, some other approaches address some aspects of these issues, e.g., Balanced Scorecard (BSC) [6] offers business goals related to four business concepts but does not explicitly provide models for use in the software domain. Practical Software Measurement (PSM) [7], conceptually similar to GQM, does not link the measurement program to business goals, as it views measurement as a project level activity.

Linking business goals to software goals must be done within a context where many different and often clashing goals, stakeholders, and factors must be taken into account. The multiplicity of environmental factors (e.g., the maturity of the software development organization, the degree of software reuse, etc.) influences how we navigate the goal space.

3. Overview of GQM+Strategies

GQM+Strategies adds several extensions on top of the GQM model. GQM represents a systematic approach for tailoring and integrating goals with models of the software processes, products, and quality perspectives of interest, based upon the specific needs of the project and the software domain of an organization.

The goals are defined in an operational, traceable way by refining them into a set of quantifiable questions that are used to extract the appropriate information from the models. The questions and models define the metrics and the metrics, in turn, specify the data that needs to be collected. The models provide a framework for interpretation of the collected data.

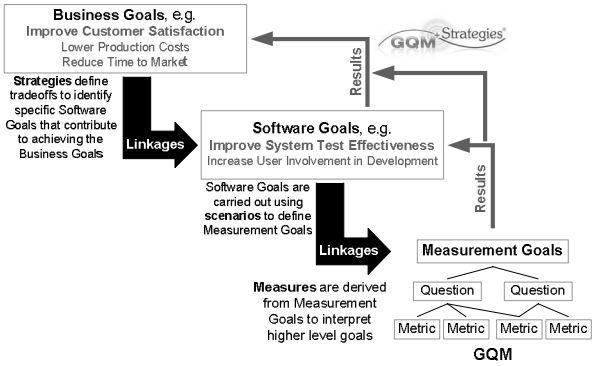

Figure 1. GQM+Strategies

In extending GQM, the GQM+Strategies approach, depicted in Figure 1, makes the *business goals*, strategies, and corresponding *software goals* explicit. *Strategies* are formulated that deal with business goals such as improving customer satisfaction, garnering market share, reducing production costs, and more, taking into account the *context* and making explicit any *assumptions*. GQM+Strategies also makes the relationships between software-related activities and *measurement goals* explicit. Sequences of activities necessary for accomplishing the goals are defined by the software organization and embedded into scenarios in order to achieve some software-related goal. Links are established between each software goal and the business-level strategy it supports. Attached to goals, strategies, and scenarios at each level of the model is information about relationships between goals, relevant context factors, and assumptions. The entire model provides an organization with a mechanism not only to

define software measurement consistent with larger, upper-level organizational concerns, but also to interpret and roll up the resulting measurement data at each level. Finally, GQM+Strategies' linkages and measures ensure the business goals are fulfilled. Figure 1 depicts the various elements of a GQM+Strategies model.

For example, assume an organization has decided to increase customer satisfaction. The strategy chosen to achieve this business goal might be to improve the reliability of the product (which includes both hardware and software elements). It is determined that the software development part of the organization could best contribute by reducing the defect slippage to the customers by improving the software system testing process. The leaders of the software unit then decide on a set of actions (i.e., a scenario) to achieve this improvement, including creating a data baseline of current defect slippage, consulting experts for potential specific improvements to the system testing process, implementing the improvements, and measuring the results. Some of these scenario steps directly generate measurement goals, e.g., measuring defect slippage. These measurement goals are then the starting point for applying traditional GQM to determining questions and specific metrics and data (e.g., system test defect data) for achieving those measurement goals.

Applying GQM+Strategies in this situation would result in an integrated model that makes the linkages explicit between the initial business goal (improving customer satisfaction) and the data generated by the software measurement plan (system test defect data). This not only provides a meaningful rationale for the software measurement effort, but also provides a blueprint for interpreting the data at each level up the chain. Assumptions may be reexamined and modified to create new strategies and scenarios. Most importantly, the effects of any changes can be understood in the context of the entire goal set.

4. Current status and next steps

Support tools are under development that take advantage of actual experiences and specific expertise by storing common business goals, strategies, scenarios, etc., and their linkages so that organizations are better able to choose and navigate through the space of options. Organizations will be able to identify their own measurement program and track it over time.

The GQM+Stratgies model is based on numerous years of experience applying GQM and understanding the need to extend the GQM model. The GQM+Stratgies model has been applied to several projects retroactively, but we have not yet applied it end-to-end to build a measurement program. We are in the process of building an experience base that incorporates multiple examples and multiple domains.

5. References

[1] V. Basili and D. Weiss, "A Methodology for Collecting Valid Software Engineering Data," IEEE Transactions on Software Engineering, vol.10 (3): 728-738, November 1984.

[2] V. Basili, G. Caldiera and D. Rombach, "The Goal Question Metrics Approach", Encyclopedia of Software Engineering, Wiley, 1994.

[3] M. Chrissis, M. Konrad, S. Shrum, "CMMI® Guidelines for Process Integration and Product Development", Addison-Wesley, 2007

[4] Office of Government Commerce (OGC): the IT Infrastructure Library (ITIL) Service Delivery, The Stationary Office London, 2002

[5] ISACA, Control Objectives for Information and related Technology (COBIT®), retrieved 04/12/2007, from www.isaca.org

[6] R. Kaplan and D. Norton, "The Balanced Scorecard--Measures That Drive Performance", Harvard Business Review, January-February 1992, p. 71.

[7] Practical Software Measurement: A Foundation for Objective Project Management, v. 4.0b, October 2000, from www.psmsc.com