DETERMINING THE IMPACT OF BUSINESS STRATEGIES USING PRINCIPLES FROM GOAL-ORIENTED MEASUREMENT

Victor Basili [1)2)], Jens Heidrich [3)], Mikael Lindvall [1)], Jürgen Münch [3)],
Carolyn Seaman [1)4)], Myrna Regardie [1)], Adam Trendowicz [3)]

Abstract
In practice, the success or failure of business strategies is often determined by management as a gut feeling without taking into account quantitative information. If data is collected, it is often unclear how the data contributes to higher-level goals of the organization. GQM$^+$Strategies® provides mechanisms for explicitly linking measurement goals to higher-level goals, and also to goals and strategies at the level of the entire business. It is based on experiences with software-related organizations, but is intended to be applicable in all kinds of businesses. This article gives an overview of the basic concepts and presents a practical case.

1. Introduction

Determining the impact of business strategies is crucial for effective decision making within an organization. Different goals and corresponding strategies exist at different levels of the overall business. In practice, these goals and strategies are not aligned and their success or failure is often determined by management as a gut feeling without taking into account quantitative information. For instance, in a software organization, engineers are frequently faced with apparently unrealistic goals related to software development. If the next version of a software product needs to be released to the market in half of the originally planned time, the software development schedule is simply cut in half. There is rarely a discussion of trade-offs or other options for such decisions in order to avoid deviations of budget and schedule. Goals and strategies need to be defined explicitly and derived from high-level business goals in a systematic and transparent way. Moreover, underlying assumptions and environmental factors are often not documented, which makes it hard to determine the reasons for failed strategies. Furthermore, if measurement data is collected on the project level, it is often unclear how the activities performed there and the data collected contribute to higher-level goals of the organization. For instance, time and money are spent on software initiatives that do, in fact, not contribute to the bottom line of the business. In practice, an approach to measurement that explicitly links high-level business goals and measurement data is needed. Building an effective measurement program is a challenging task in itself. It involves observation, experience facilitation, collaboration, decision making, analysis, and synthesis regarding goals, context factors, and assumptions. Furthermore, it assumes an organizational structure that sustains the process and learns. Most organizations fall short of putting together a successful program. This article presents the GQM$^+$Strategies® approach [2] for explicitly linking measurement goals to

[1] Fraunhofer Center for Experimental Software Engineering, Maryland College Park, Maryland 20742, USA

[2] University of Maryland, Maryland College Park, Maryland 20742, USA

[3] Fraunhofer Institute for Experimental Software Engineering, Fraunhofer-Platz 1, 67663 Kaiserslautern, Germany

[4] University of Maryland, Baltimore County, Maryland 21250, USA

higher-level goals for the organization, and also to goals and strategies at the level of the overall business. Even though the development of the approach was focused on software-related organizations, the basic concepts can be generalized to set up an organization-wide measurement program for controlling business strategies. GQM$^+$Strategies® is based on the familiar Goal Question Metric paradigm [1], which is in widespread use today for creating and establishing measurement programs throughout the software industry. This 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 and strategies. Section 2 gives a detailed step-by-step description of how to systematically connect goals on an organization's different levels of abstraction using the GQM$^+$Strategies® approach. Section 3 presents an example case of how to apply the approach, Section 4 briefly illustrates practical benefits, Section 5 addresses related work, and Section 6 presents final conclusions and future work.

2. Basics of the GQM$^+$Strategies® Approach

Based on the authors' experience with establishing measurement programs, we see three essential factors for putting together an effective and sustainable measurement program. The first factor is to define the right goals. This includes linking goals at all logical levels of the organization (e.g., business, system, software, and project), using corporate goals to generate lower-level goals, and inheriting lower-level goals from upper-level goals. Moreover, it is important to identify the context and temporal aspects of a goal, such as what are we focusing on, or what shall be achieved and by when. The second factor is to collect the right data; that is, quantifying and interpreting the goals at all levels, justifying what data is collected at each level and why, maximizing the benefits, as well as minimizing data collection and data analysis costs. This also includes taking maximum advantage of already existing data (instead of defining everything from scratch in a pure top-down manner). The third factor is to define and sustain the measurement process. This includes creating the right organizational structure, getting feedback to projects in a timely fashion, and maintaining commitment within *all* organizational levels (from top-level management down to project managers).

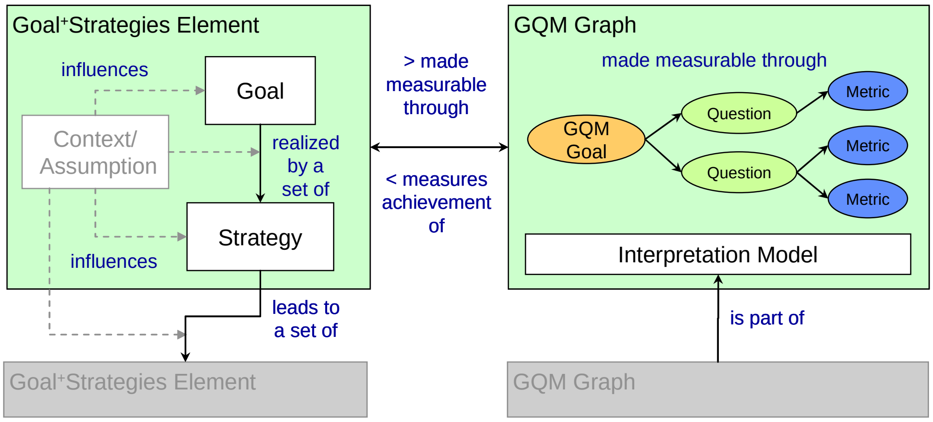

Figure 1: Meta-Model Elements for Constructing a GQM$^+$Strategies® Grid

2.1. The Conceptual Model

GQM$^+$Strategies® [2] is an approach for clarifying and harmonizing goals and strategies across all levels of an organization, communicating business goals throughout the whole company, aligning goals with strategies, monitoring the deployment strategy, and obtaining feedback about the success or failure of strategies and business goals. One central component of the GQM$^+$Strategies$^®$ approach is the conceptual model: The GQM$^+$Strategies$^®$ *grid* specifies goals and strategies across all levels of an organization including the measurement program needed to monitor and control them. Figure 1 gives an overview of all conceptual elements for constructing a grid. The meta-model allows multiple goal levels and permits deriving multiple strategies for each of these goal levels. A goal may be realized by a set of strategies, which may in turn lead to a sequence of goals. A set of predefined goals and strategies may be defined as part of an (organization-specific) experience base. Selection and adaptation of predefined goals and strategies as well as definition of new goals and strategies is driven by context factors and assumptions. *Context factors* are environmental variables that represent the organizational environment and affect the kind of models and data that can be used. *Assumptions* are estimated unknowns that can affect the interpretation of the data. The entire model provides an organization with a mechanism not only for defining measurement consistent with larger, upper-level organizational concerns, but also for interpreting and rolling up the resulting measurement data at each level. At each goal level, measurement plans are defined in order the measure the achievement of the defined goal in combination with the chosen strategy. This includes the definition of GQM measurement goals, the derivation of questions and metrics, as well as the definition of an interpretation model that determines whether the measurement goal has been achieved. A single GQM goal (that measures a GQM$^+$Strategies$^®$ element), corresponding questions, metrics and interpretation models are called a *GQM graph* in the GQM$^+$Strategies$^®$ notation.

2.2. The Grid Derivation Process

The GQM$^+$Strategies$^®$ grid derivation process [2] consists of a couple of basic tasks that may be instantiated depending on the concrete application scenario. For instance, if you want to start with a software-related goal and want to relate upper-level business goals as well as lower-level project goals to it, you need a different sequence of tasks than if you want to derive the whole grid in a top-down-oriented manner. For simplicity reasons, we present an activity diagram of the top-down process in Figure 2. There are basically two different kinds of processes, which may be performed in parallel. On the left side, there are tasks needed to define goals and related strategies. This process is iterated as long as all organizational levels (for which the measurement program has to be created) are defined. On the right side, there are tasks needed to measure already defined goals and strategies; that is, corresponding measurement goals are defined and questions, metrics, and interpretation models are specified using the GQM approach. This measurement branch can be built up in parallel to breaking down goals and strategies. The latter process consists of the following tasks:

- *Elicit General Context & Assumptions:* Before defining business goals for an organization, the basic motivation needs to be determined and the rationales that lead to defining the goal need to be described. (1) First, context factors of the organization are defined. This includes characterizing the product or service provided, identifying existing processes, tools, etc., characterizing typical customers, characterizing income sources and business model (e.g., characterizing the factors that influence profitability, contracting vehicles, etc.), characterizing organizational interfaces (such as levels in the overall organization and levels in the contract chain), and finally, characterizing existing measurement programs and already collected data. (2) Second, underlying assumptions that lead to business goals are documented. For instance, this includes what one believes to be true but has little or no empirical evidence about, as well as assumptions about the technology, market, customers, the organization, or the workforce.

- *Define Top-Level Goals:* (1) First, an initial set of potential high-level goals has to be identified. These goals may be derived from asking some basic questions, such as: What are the organizational principles that you do not want to change, i.e., aspects of the organization you want to keep as is? What are the key elements of the environment (such as transparency, employee satisfaction, controlled risk, learning environment)? Is the organization risk-averse, risk-neutral, or risk-driven? What are your existing business goals? What do you want to happen next? Where do you envisage your organization in 5 or 10 years? How do you want to grow, e.g., new customers, new competencies? How would you define success, e.g., do you want to improve some aspect of the business? (2) Second, once a list of initial goals has been defined, the goals have to be prioritized. This also includes identifying conflicts and other relationships between the goals. (3) Third, once the most important goals have been determined, they need to be formalized using the GQM$^+$Strategies$^®$ goal template. The template documents the basic *activity* that should be performed in order to accomplish the goal, the main (quality) *focus* of the goal, the *object* under consideration (e.g., a process or product), the *quantification* of the goal specified by a magnitude, the *timeframe* in which the magnitude has to be achieved, the *scope*, and basic *constraints* that may limit accomplishing the goal. Furthermore, potential *relationships* with other (complementary or competing) goals are listed.

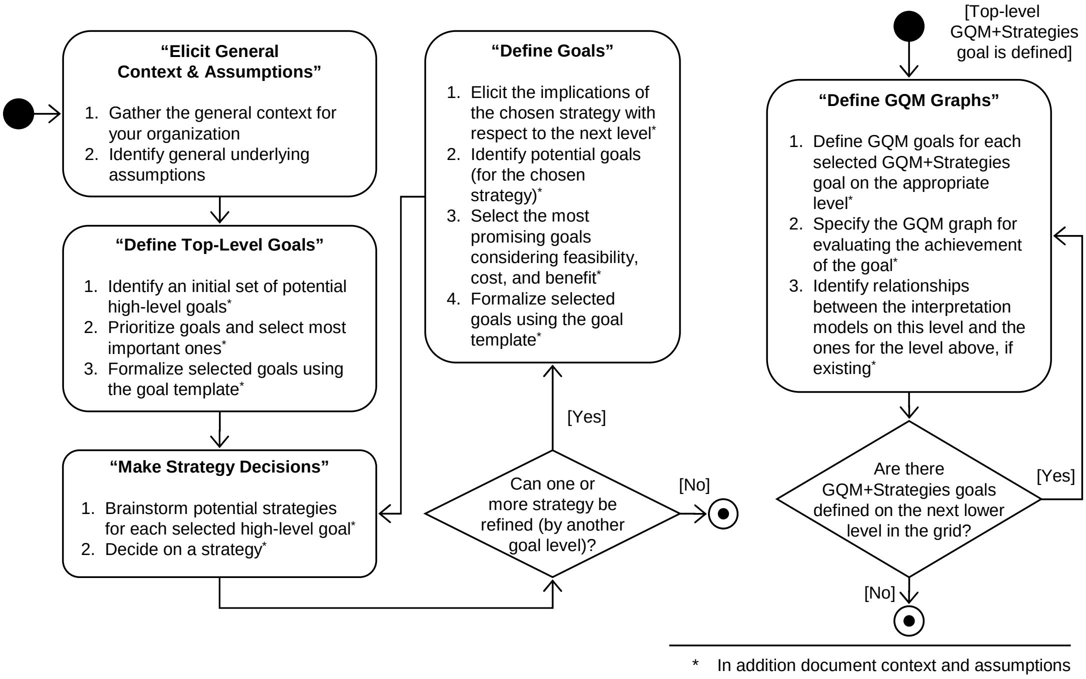
"Elicit General Context & Assumptions"
1. Gather the general context for your organization
2. Identify general underlying assumptions
"Define Top-Level Goals"
1. Identify an initial set of potential high-level goals*
2. Prioritize goals and select most important ones*
3. Formalize selected goals using the goal template*
"Make Strategy Decisions"
1. Brainstorm potential strategies for each selected high-level goal*
2. Decide on a strategy*
"Define Goals"
1. Elicit the implications of the chosen strategy with respect to the next level*
2. Identify potential goals (for the chosen strategy)*
3. Select the most promising goals considering feasibility, cost, and benefit*
4. Formalize selected goals using the goal template*
[Yes]
Can one or more strategy be refined (by another goal level)?
[No]
[Top-level GQM+Strategies goal is defined]
"Define GQM Graphs"
1. Define GQM goals for each selected GQM+Strategies goal on the appropriate level*
2. Specify the GQM graph for evaluating the achievement of the goal*
3. Identify relationships between the interpretation models on this level and the ones for the level above, if existing*
Are there GQM+Strategies goals defined on the next lower level in the grid?
[Yes]
[No]
* In addition document context and assumptions

Figure 2: The GQM$^+$Strategies$^®$ Grid Derivation Process

- *Make Strategy Decisions:* (1) First, a list of potential strategies for achieving the business goal needs to be identified (e.g., building a more reliable system that will lead to less customer complaints vs. testing reliability in). (2) Second, the most promising strategy has to be selected considering its cost and benefit, and taking into account context factors and assumptions that naturally restrict the feasibility of potential strategies.

- *Define Goals:* This task is only conducted if one or more strategies could be refined by another goal level. The stop criterion depends on the organization for which the GQM$^+$Strategies$^®$ model is built. Usually, the process stops if a concrete set of steps has been derived that may easily be implemented in the organization (e.g., on the project level). (1) First, the implications of the chosen upper-level strategies (e.g., strategies of the business level) with respect to lower-level goals (e.g., software development-specific goals) have to be elicited. For instance, if a chosen strategy is to test in reliability, the software test processes must be examined. (2) Second, potential lower-level goals need to be identified based on this analysis (e.g., decrease customer reported defects by improving system test effectiveness). (3) Third, the most promising goal with respect to feasibility, cost, and benefit is selected. Again, context factors and assumptions help to define the right selection criteria. (4) Fourth, the lower-level goal needs to be formalized using the goal template as described above (see "*Define Top-Level Goals"*).

Creating the measurement branch of the grid for each goal and strategy level is not an isolated task; that is, the metrics derived across different levels of the GQM$^+$Strategies® model will usually overlap. Moreover, an interpretation model for a higher-level goal may only be defined completely if the lower-level pieces have already been modeled.

Define GQM Graphs: A measurement plan has to be developed for evaluating the achievement of goals and strategies. (1) First, potential measurement goals need to be identified. The GQM goal template [2] (object, purpose, quality aspect, viewpoint, and context) is used for formalizing the measurement goal. (2) Second, GQM questions and metrics as well as criteria for evaluating the achievement of the measurement goals are determined and interpretation models are defined for aggregating and interpreting the collected measurement data. With that, the GQM$^+$Strategies® model provides guidance not just for planning, but also for analyzing and rolling up the resulting data for the decision makers and helps to make the right decisions for achieving the designated goal and evaluating the implementation strategy. (3) Third, relationships between the interpretation models on this level and the one above need to be identified. The reason for that is that using the information from the current level may help to extend and improve the upper-level interpretation models significantly.

3. Applying the GQM$^+$Strategies® Approach

The example case presented in this section is a hypothetical project, built upon real project experience gained over many years of industrial work. GQM$^+$Strategies® was applied post mortem in order to evaluate the applicability of the conceptual model. In the following, the activities that were performed in order to systematically define the measurement program according to the GQM$^+$Strategies® method are described. The two branches of the grid derivation process presented in the previous section are merged into a continuous series of tasks.

Elicit General Context & Assumptions: A company called Company X creates web applications in the financial area. Their most famous product is called Splash and Company X is the market leader for this class of product. The basic motivation for defining a measurement program is that the market for this class of product is becoming highly competitive and there is a need to safeguard the place in the market.

Table 1: Formalized Goal on the Business Level

Activity	Increase
Focus	Customer satisfaction
Object	Product "Splash"
Magnitude	10% reduction in number of customer complaints
Timeframe	12 weeks after release
Scope	Web Products Division, Splash Project Manager
Constraints	Splash price and functionality
Relations	Can conflict with development cost goals, schedule goals, …

Define Top-Level Goals: A way to safeguard the place in the market is to keep existing customers by generating customer loyalty. This can be achieved by improving customer satisfaction with the next product. Other potential high-level goals were not defined in our case. However, the organization could also think about decreasing time to market in order to speed up delivery of its product to the customer. The business goal "increase customer satisfaction for the next product" is selected as the only goal and is formalized according to the goal template. The results can be seen in Table 1.

Define Top-Level GQM Graphs: The GQM goal for evaluating the defined business goal is as follows: Analyze customer complaints trend for Splash for the purpose of evaluation with respect to a 10% improvement over history from the point of view of quality management in the context of the Web Products Division of Company X. Questions and metrics are derived and an interpretation model for evaluating the achievement of the goal is defined: If the ratio between customer complaints for Splash and a historical baseline over a 12-week time period is less than or equal to 0.9, the business goal is achieved.

Make Strategy Decisions (on Business Level): An analysis revealed that many customer complaints are due to product reliability problems. Brainstorming potential strategies resulted in the following alternatives: (S1) build reliability in (e.g., implement fewer defects) or (S2) test reliability in (e.g., remove more defects). Company X has only little control over the development process and only a limited budget for process improvement. Thus, they decide to "test reliability in".

Define Goals (on Software Level): First, we elicit the implications of the chosen strategy with respect to software development. In order to test reliability in, the software test processes must be examined. Potential software goals include decreasing customer-reported defects by improving (G1) system test effectiveness, (G2) unit test effectiveness, or (G3) acceptance test effectiveness. Company X has discovered a new system test process that seems appropriate for their context. They assume (based on their experience) that they can decrease the total number of customer complaints by 10% by reducing customer-reported software field defects (i.e., those that slip by system test) by 20%. So the most promising goal is to decrease customer-reported defects by improving system test effectiveness. A formalization of the goal is presented in Table 2.

Table 2: Formalized Goal on the Software Level

Activity	Decrease
Focus	Customer-reported software defects
Object	System test process for Splash
Magnitude	20%
Timeframe	12 weeks after release (might check every week)
Scope	Web Products Division, Splash Software Manager
Constraints	Development cost and functionality
Relations	Can conflict with development cost goals, schedule goals, …

Define GQM Graphs (on Software Level): The corresponding GQM goal is to analyze the trend in unique customer complaints that are due to software defects for the purpose of evaluation with respect to a 20% reduction when compared to prior projects from the point of view of quality management in the context of the Web Products Division of Company X. The goal is achieved if the number of unique customer complaints that are due to software defects of Splash compared to the average number of complaints of a set of baseline products is less than or equal to 0.8. This new information has some implications for the interpretation model of the business goal. For example, if the number of customer complaints is reduced by 10% or more, then we have achieved our business goal, else – if the number of unique customer complaints due to defects is reduced by 20% – the assumption may be wrong. It is necessary to consider that perhaps the unique customer complaints due to defects are not a major problem relative to other customer complaints. It is then necessary to identify the major sources of complaints and redefine the strategy or reconsider the business goal.

Make Strategy Decisions (on Software Level): Because there is a new system test process that seems appropriate for Company X, the one and only strategy is to introduce the new system test process.

Table 3: Formalized Goal on the Project Level

Activity	Decrease
Focus	Defect slippage
Object	New system test process for Splash
Magnitude	20%
Timeframe	12 weeks after release (might check every week)
Scope	Web Products Division, Splash Software Manager
Constraints	Development cost and functionality
Relations	Can conflict with development cost goals, schedule goals, …

Define Goals (on Project Level): Company X assumes that reducing slippage by 20% reduces reported defects by 20%. They already collect defect slippage data and have created a corresponding baseline based on the assumption that the projects that form the baseline are relevant to the current Splash development project. The goal is to apply the new system test method in order to see if it reduces defect slippage by at least 20% and generates the necessary improvement to customer complaints. The goal is formalized in Table 3.

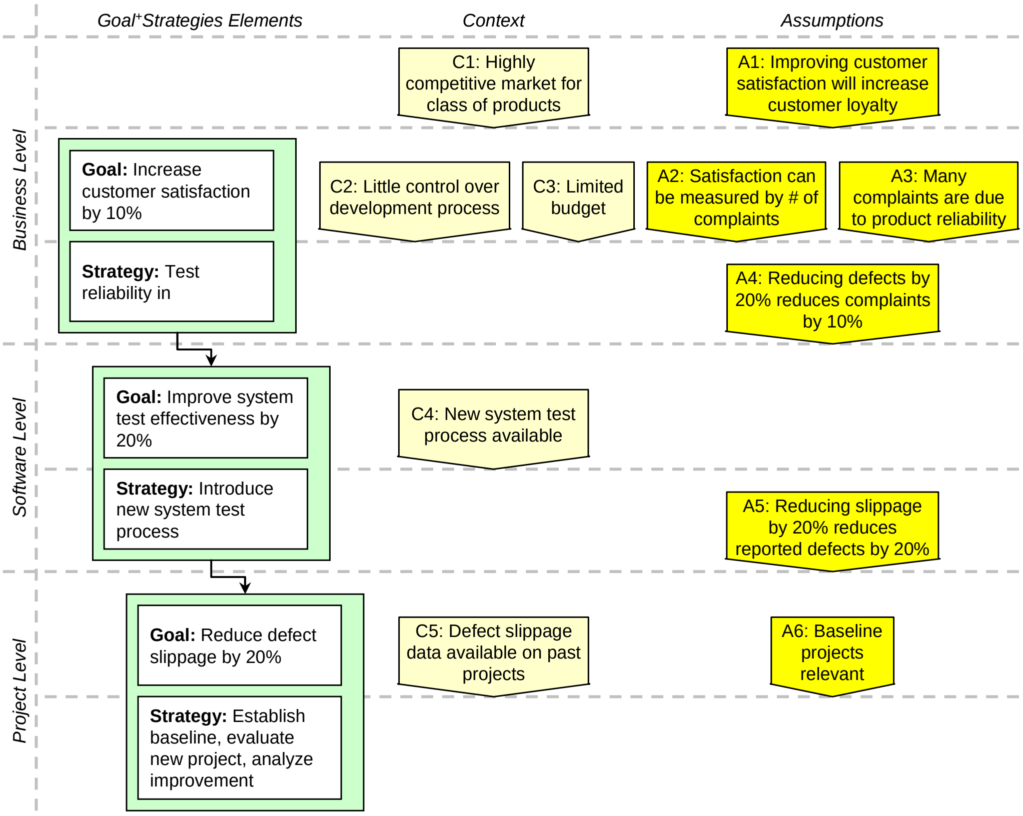
Goal+Strategies Elements
Context
Assumptions
Business Level
Software Level
Project Level
C1: Highly competitive market for class of products
A1: Improving customer satisfaction will increase customer loyalty
Goal: Increase customer satisfaction by 10%
Strategy: Test reliability in
C2: Little control over development process
C3: Limited budget
A2: Satisfaction can be measured by # of complaints
A3: Many complaints are due to product reliability
A4: Reducing defects by 20% reduces complaints by 10%
Goal: Improve system test effectiveness by 20%
Strategy: Introduce new system test process
C4: New system test process available
A5: Reducing slippage by 20% reduces reported defects by 20%
Goal: Reduce defect slippage by 20%
Strategy: Establish baseline, evaluate new project, analyze improvement
C5: Defect slippage data available on past projects
A6: Baseline projects relevant

Figure 3: Summary of Context Factors and Assumptions

Define GQM Graphs (on Project Level): The GQM goal for evaluating the goal on the project level is to analyze the system test process for Splash for the purpose of evaluation with respect to 20% defect slippage compared to prior projects from the point of view of quality management in the context of the Web Products Division of Company X. The goal is achieved if the ratio of faults found in system test to that found after system test on the Splash project compared to the same average ratio in the set of baseline projects is greater than or equal to 1.2. If the value is between 1 and 1.2, then the new method is better than history but not good enough to achieve the goal. The relationships between this interpretation model and the one for the software goal are as follows: If the customer-reported defects are reduced by 20%, then we have achieved our software goal, else –

if defect slippage from system test is reduced by at least 20% – one of the previous assumptions is wrong (either reducing slippage by 20% does not reduce reported defects by 20% or the projects that form the baseline are not relevant to the current project). There are more GQM goals (not discussed in this example) on this level that could be followed depending on whether historical data on defect slippage exists. However, because of space limitations, we skip the remaining goals. A comprehensive overview of the goals, strategies, and related context factors and assumptions that helped in going from one level to the other can be found in Figure 3.

4. Practical Benefits

In practice, multiple goals on the same level of abstraction or on different levels of the GQM$^+$Strategies$^®$ model will most likely be interrelated with each other. The most obvious relationship (documented in the conceptual model) is a hierarchical structure consisting of a top goal and sub-goals connected via strategies. For a certain goal, a set of complementary goals may exist, which provide additional support for the current goal. On the other hand, a set of competing goals may also exist that conflict with the current goal, whereas other goals may be totally unaffected by the current goal. There are several approaches available in practice and research to resolve goal conflicts (e.g., the Model-Clash Spiderweb diagrams [4]). However, sometimes it is hard to determine whether a strategy selected for one goal is in conflict with another goal. For instance, if our goal is to "increase product quality" and the derived strategy is "find more defects", then the strategy may have various effects on a goal like "increase profits". On the one hand, costs are increased by putting more effort into finding defects. On the other hand, costs may also be decreased, because rework effort may be reduced. GQM$^+$Strategies$^®$ allows us to explicitly model potential conflicts and also model assumptions we made in order to overcome potential conflicts. If a goal could not be achieved, interpretation models help to determine which parts of the model need to be changed and how to measure the effect of these changes. The GQM$^+$Strategies$^®$ approach makes high-level goals, strategies, and related measurement goals explicit across all levels of an organization. The entire model provides an organization with a mechanism not only for defining software measurement consistent with larger, upper-level organizational concerns, but also for interpreting and rolling up the resulting measurement data at each level. However, in order to implement a sustainable measurement program, further activities have to be performed (which are part of the overall approach, but beyond the scope of this paper), including (a) measurement instrumentation, (b) data collection and validation, (c) data analysis, visualization, and interpretation, and (d) the practical introduction and management of a measurement program.

5. Related Work

CoBIT (Control Objectives for Information and related Technology) [6] is a framework that focuses on controlling IT. CoBIT was initially developed by the Information Systems Audit and Control Association (ISACA) and has been developed by the IT Governance Institute since 2000. CoBIT distinguishes between four different levels of goals: business goals, IT goals, process goals, and activity goals. The list of goals provided represents the consensus of experts. CoBIT provides metrics and models to measure their achievement. The CoBIT process model subdivides IT into four domains and 34 processes in line with the responsibility areas of plan, build, run, and monitor IT processes. It has a clear focus on IT services, provides an easy structure for documenting relationships between and among goals and metrics, and summarizes best practices for IT governance. However, it provides only a fixed set of goals, has limited tailoring support, and makes assumptions about the process (reference model) used. Moreover, it provides no means for documenting underlying context information and assumptions that would help to tailor the model to specific organizations.

Practical Software and Systems Measurement (PSM) [8] defines an information-driven measurement process that guides project managers in selecting, collecting, defining, analyzing, and reporting specific software issues. PSM represents best practices used by measurement experts. It is sponsored by the Department of Defense and the US Army. It focuses on measurement on the project level and defines seven common areas including schedule, cost, size, product quality, performance, effectiveness, and customer satisfaction. The areas may be extended with respect to specific goals related to a project. The method is tool-supported and comes with a set of predefined goals and metrics that may be tailored to specific needs. It provides support for multiple measurement areas, defines corresponding metrics on the project level, and provides best practices for implementing a measurement program on the project level. However, it does not address measurement on the business level and does not link measurement on the project level to higher-level business goals and strategies.

Balanced Scorecard (BSC) [7] is an approach for measuring whether the activities of a company meet its goals with respect to vision and strategy. It was initially developed by Robert S. Kaplan and David P. Norton and is used by decision-makers for controlling organization-wide goals, making use of different dimensions that have to be determined based on the goals and characteristics of an organization. In order to derive goals and document relationships among them, a strategy map is defined, which connects all dimensions by using causal links (also known as BSC stories). Typically, four different dimensions are controlled (learning & growth, internal process, customer, and financial), each including one to two business goals. For each goal of a BSC dimension, corresponding indicators, target values, and initiatives (activities) are defined. BSC helps to make goals on the business level explicit, defines measures for controlling the goals, and sets target values. It helps to define causal chains for strategies in order to achieve business goals. However, depending on the organization, it is hard to formulate goals (no template is provided) and there is no consistent linkage between measurement on the business level and on the project level. Becker and Bostelman [3] address the misalignment between strategy at the organizational level and the project level of software organizations caused by project data that does not address organizational goals and organizational goals that fail because they are not operationalized through processes and metrics at the project level. Buglione and Abran [5] also discuss the misalignment problem, but their work is more of a comparison, rather than a merger, of BSC and GQM. The M^{3}P – Model, Measure, Manage Paradigm [9] – is an extension of the QIP and GQM. M^{3}P embeds GQM as a measurement definition technique within a larger framework that encompasses organizational concerns, but does not allow for explicitly linked goals at different levels of the organization.

Six Sigma (6σ) [10] is a set of practices designed to improve manufacturing and business processes and seeks to identify and eliminate the causes of defects. It was first defined by Bill Smith at Motorola in 1986 and can be used for creating new or controlling existing manufacturing or business processes in order to deliver products or services that meet customer needs. Six Sigma asserts that continuous efforts to achieve stable and predictable process results (i.e., to reduce process variation) are of vital importance to business success. It adapts the Plan-Do-Check-Act paradigm for continuous improvement. It utilizes tools and techniques for fixing problems in manufacturing and business processes in a continuous and disciplined fashion and focuses on decision making based on facts and data rather than on assumptions. However, it requires large amounts of reliable measurement data, which, in contrast to manufacturing, are typically not available in the software development domain. The sigma-level is assumed to be the major factor contributing to the fulfillment of customer needs.

6. Summary and Conclusions

This paper presented the GQM$^+$Strategies$^®$ approach, which provides guidance not just for planning, but also for analyzing and rolling up the resulting data for the decision-makers. The approach helps to make the right decisions for achieving the designated business goals and evaluating the implementation strategy. GQM$^+$Strategies® inherits GQM's benefit of assuring that the metrics set is as small as possible and that the data collected address the defined organizational objectives. It extends GQM by providing explicit support for linking software measurement goals to organizational business objectives. Considering this linkage is essential for organizational success, as it helps to translate strategic-level objectives into a set of operational software goals and corresponding quantitative project management. It helps to justify software measurement efforts and allows measurement data to contribute to higher-level decisions. GQM$^+$Strategies® provides a systematic way to deal with relationships between different objectives at various organizational levels. Early identification of conflicting objectives may help to prevent failures very early, namely, at the time of defining the organizational strategy. Moreover, a transparent way of specifying and synchronizing objectives at various operational levels supports understanding of the overall business and improves communicating goals and strategies. The GQM$^+$Strategies® development team is currently setting up a set of pilot projects for evaluating the approach in practice. Furthermore, tool support for specifying a GQM$^+$Strategies® grid is under development.

7. Acknowledgements

This work was supported in part by the German Federal Ministry of Education and Research (QuaMoCo Project, No. 01IS08023C) and the Stiftung Rheinland-Pfalz für Innovation (Q-VISIT Project). Furthermore, we would like to thank Sonnhild Namingha for reviewing the article.

8. References

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

[2] BASILI, V.R., et al., A., Bridging the Gap Between Business Strategy and Software Development, In: Proceedings of the International Conference on Information Systems (ICIS), December 9-12, 2007, Montréal, Canada.

[3] BECKER, S.A., BOSTELMAN, M.L., Aligning Strategic and Project Measurement Systems, IEEE Software, May/June 1999, pp. 46-51.

[4] BOEHM, B., PORT, D., AL-SAID, M., Avoiding the Software Model-Clash Spiderweb, Computer, vol. 33, no. 11, pp. 120-122, Nov., 2000.

[5] BUGLIONE L., ABRAN A., Balanced Scorecards and GQM: What are the Differences?, In: Proceedings of FESMA-AEMES Software Measurement Conference, October 2000.

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

[7] KAPLAN, R, NORTON, D., The Balanced Scorecard - Translating Strategy into Action, Boston: Harvard Business School Press, 1996; ISBN 0-87584-651-3.

[8] MCGARRY, J., CARD, D., JONES, C., LAYMAN, B., CLARK, E., DEAN, J., HALL, F., Practical Software Measurement: Objective Information for Decision Makers, Addison-Wesley Professional, 2001.

[9] OFFEN, R.J., JEFFEREY, R., Establishing Software Measurement Programs, IEEE Software, Mar/Apr 1997.

[10] PYZDEK, T., The Six Sigma Handbook: A Complete Guide for Green Belts, Black Belts and Managers at All Levels. New York, NY: McGraw-Hill Professional, 2003.