



Understanding Software Measurement, and How to Apply it for Your Organization

Daniel Horvath

INTRODUCTION

Software development and maintenance in an organization requires an increasingly larger budget of time, resources and money. In the context of this paper, an organization may be an entire corporation, a division or other sub-group within a larger company. It is further assumed that the organization is of a medium to large size; large enough to have a significant expenditure on information technology support. It behooves every such organization to do everything possible and practical to control these critical processes and resources. The method for such control is a software measurement program.

Software measurement enables organizations to gain an understanding of where and how the expenditures of time, resources and monetary funds are being spent on software development and maintenance activities. Only with this understanding is it possible to gain control over these activities and to make any improvements. This control as well as the recognized and implemented improvements to processes, tools and resources, will enable greater efficiency and cost efficacy.

This article is intended to provide a basic understanding of Software Measurement and how it may be implemented in an organization. Most of the information in this article is based on the *Guidelines to Software Measurement*,

Release 2 publication. Other sources were used and are noted when referenced. *Guidelines to*

Software Measurement is available from the International Function Point Users Group, at www.ifpug.org/publications/.

SOFTWARE MEASUREMENT BASICS

Guidelines to Software Measurement defines Measurement as the assignment of a relative value. Metrics are combinations of attributes or measures from this activity. One commonly used metric involves dividing effort amount (hours, days or months) by function points to yield a productivity rate. Another well known metric is the division of the number of defects by function points to yield defect density rate.

Practical Software Measurement (PSM) is a measurement framework that has been developed to be used for software development, and also Systems engineering, process improvement and measurement at the enterprise level. Project information is tracked in terms of issues, concerns, risks and problems. It is then mapped and prioritized into one of seven information categories:

- Schedule and Progress
- Resources and Cost
- Product Size and Stability
- Product Quality

- Process Performance
- Technology Effectiveness
- Customer Satisfaction

Practical Software Measurement maps these categories further into measureable concepts, and then to specific measures. The entire PSM model is presented in a Category Concept Measure table that includes 61 measures.

PSM information is available at the PSM Support Center Website: www.psmcsc.com.

Goal Question Metric (GSM) is another measurement framework. It allows users of the methodology to tie measurements to goals.

There are three steps:

1. Identify goals of the organization. This may be done by interviewing the leaders of the organization. An example may be: “reduce defects”.
2. Formulate questions for each goal. Examples may be: “how many defects are there?”, and: “what is the defect density?”
3. Analyze the questions to determine if there is a quantitative way to measure them. For the above examples, we may want to use the total number of defects over a specific period of time (for instance a month) and function point size of the application. The defect density (number of defects divided by the number of function points) provides the answer to the questions.

GSM is effective because it keeps the focus on not only the metrics, but more importantly, the organization’s goals.

Both PSM and GSM were used to develop the Measurement and Analysis Process Area of the

Capability Maturity Model – Integrated (CMMI). CMMI is generally regarded as the most well-known process improvement framework for software development. In CMMI, measurement is part of the foundation for process improvement. More information about CMMI is available at the Software Engineering Institute website: <http://www.sei.cmu.edu/cmmi/>. Additional information is available at the Q/P Management Group website: <http://qpmg.com/>. Note that Q/P’s Center of Excellence (COE) provides a range of Software Engineering Institute (SEI) CMMI-based and proprietary process improvement assessments, training, and support for implementation of recommendations.

Regardless of methodology or framework, most measurement programs include the following fundamental measures:

- Size
- Defects
- Effort
- Duration
- Cost
- Customer Satisfaction

More information on these measures is available in the *Guidelines to Software Measurement*. With this understanding of the basics, we can now present further information about how to implement and sustain a Software Measurement Program.

STAKEHOLDERS AND PARTICIPANTS

One of the first steps toward implementing a software measurement program is to identify the stakeholders and participants. These are the people who are the most involved with the process and they may also be the audience and beneficiaries of it. They will also be involved as the measurement program is sustained later in its life cycle.

- **Process Owner:** The person who is responsible for establishing the framework for the business process.
- **Process Manager:** The person who is responsible for the operation and management of the process framework, based on guidance from the Process Owner.
- **Process Members:** The people involved with the product or service development for the process owner.
- **Measurement Facilitators:** Internal or external resources, which are used in the measurement process to encourage participant involvement, identify and explore areas of concern, and ensure that all possible feedback and input is included.
- **Project Managers and Business User Representatives:** As the people responsible for the time, budget and quality of software projects, these individuals from the IT and User sides, are both involved with the gathering of measurements and with receiving the benefits resulting from the measurements.
- **Information Technology Managers and Business Line Managers:** On the IT side, these people have the responsibility to install the infrastructure and to supply resources needed for the measurement process effort. On the business side, they are interested in achievement of business objectives and the satisfaction of customers. In general these managers

provide resources for the process and also benefit directly from it.

- **Executive Managers:** Executive managers set goals for the organization. A person from this group may be the process owner.

FEATURES OF A SOFTWARE MEASUREMENT PROGRAM

The following features are components in effective software measurement programs.

- **Alignment with business objectives.** This enables prioritization of effort and resources as well as better communication with customers.
- **Balanced Set of Metrics.** This feature provides a comprehensive perspective and helps establish a baseline for the determination of trends.
- **Decision Process Information.** Measurement information is used for management and estimation as part of the decision making process.
- **Identified areas for Action Plans.** This feature allows measurement to provide information about Continuous Process Improvement (CPI) plans and includes cost savings, increased quality and improved productivity.
- **Integration of Measurement into development and support processes.** Measurement tasks are part of the software development and maintenance processes for CPI.
- **Reporting of Measurement Results.** This feature provides meaningful information about the results gleaned from the measurement process.

- **Standardized and documented audit process.** This feature helps to ensure that accurate, consistent and repeatable data is collected and maintained for the organization.

IMPLEMENTATION OF A SOFTWARE MEASUREMENT PROGRAM

It is recommended that a project-based approach be used to implement a Software Measurement Program. The components of the project plan should include planning, analysis, implementation & measurement and evaluation & improvement. The evaluation and improvement step can also be considered part of the effort to sustain the program for the organization. Each of these components is made up of the activities outlined below.

Planning: Definition of Goals, Objectives, Benefits

The most important step in establishing a software measurement program is to define the goals, objectives and benefits. This determines the basis for what is to be measured. The Process Owner, Process Manager, and Executive Managers are involved in this activity.

Planning: Establishment of Sponsorship

Business and IT Managers, the Process Owner, Process Manager, as well as Project Managers and Process Members all participate in this activity. Supportive sponsorship is crucial to the success of any metrics program. Sponsorship enables the effort to be understood as a priority for the organization. A critical success factor for the measurement program is to ensure that an executive sponsor is identified and actively participates in the development of the program.

Planning: Communication and Promotion of Measurement

The measurement program must be sold to the organization. This is mainly the responsibility of

the Process Manager and the Process Members. The program is sold in order to create further understanding of the priority as well as the goals of the program.

Planning: Identification of Roles and Responsibilities

This step may be done in parallel with the next one: Analysis of Audience and Identification of Target Metrics. Here, the roles and responsibilities of participating individuals are defined and communicated to the team and the organization.

Analysis: Analysis of Audience and Identification of Target Metrics

Audience analysis is conducted in order to ensure that the appropriate data is measured and tracked. From this analysis the target metrics can be derived and understood. An example would be the designation of Project Managers as the Audience, and in this case, the identification of Productivity Rate as a Target Metric.

Analysis: Definition of Software Measures

After the target metrics have been identified, they must be further defined and developed. All measures that make up the metrics must be clearly understood. This is important to help ensure that the correct data is collected and that it is done consistently. For example, if the goal is to determine whether productivity is improving, detailed data relating to project size and effort will need to be collected.

Analysis: Definition of the Data Collection, Analysis and Data Storage Approach

How data is collected and tracked depends on what data is required. Data collection tools and repositories may be required for large efforts. In addition, standards and processes for this effort need to be defined and developed. The approach selected is ultimately based on organizational goals and constraints. This activity is the responsibility of the Metrics Process Members Team.

Implementation & Measurement: Education

Education as to why measurement is necessary, how it affects individuals in the organization, how it can help, and individual responsibilities are crucial to the success of the implementation. Training should be tailored to different groups as needed. One example of a group with a critical training need is the Software Measurement Coordinators. This training would include: guidance in data collection and analysis techniques, consulting skills, and instruction about how to bring about how to effectively introduce measurement to the organization.

Implementation & Measurement: Reporting and Publishing Results

The measurement results should generally include many of the following: executive summary, an analysis/explanation, key statistics, simple graphs, information sources. *Guidelines to Software Measurement* contains further information about reporting.

Evaluation & Improvement: Managing Expectations

The management of expectations is the responsibility of the Process Owner and Process Manager. It must be done throughout the lifecycle of the measurement program.

Evaluation & Improvement: Managing with Metrics

This is where the measurement information is actually used and the program begins to pay for itself. The people who developed and

implemented the metrics program are not necessarily the ones responsible for using the information to make management decisions. Decision makers receive input from the metrics program, but the decisions themselves are outside the scope of the metrics program itself. The metrics team may be able to help by providing further analysis of the reporting, as necessary.

CONCLUSION

This article has provided some basic information and food for thought regarding software measurement. More specifically, it has shown how such measurement may be implemented and used in an organization. The choice of which actual metrics are to be collected and analyzed have been mentioned as examples in this document; much more thought would be necessary to determine what to measure and how to accomplish it. Most importantly, it must be noted that a Software Measurement Program, once implemented, must be *sustained*. The sustenance must be planned for as part of the initial planning and implementation activities.

Q/P Management Group is highly experienced and poised to assist organizations in all aspects of the implementation of a Software Measurement Program. For more information about how to get started, please refer to the Q/P website: <http://qpmg.com/>.

About the author

Daniel Horvath, Senior Management Consultant, Q/P Management Group wrote this article. Mr. Horvath specializes in project management and software engineering metrics, including function point analysis.