MEASUREMENT IN SOFTWARE DEVELOPMENT PROCESS

Lubomír KUBÁT, Doctoral Degree Programme (1)
Dept. of Information Systems, FIT, BUT
E-mail: kubat@fit.vutbr.cz

Supervised by: Dr. Jitka Kreslíková

ABSTRACT

The paper presents the possibilities of modeling metric system for software development and systems engineering processes. Metrics can help to evaluate and improve the quality of Software Life Cycle Processes. Relationship metrics models to quality management and quality assurance standards are mentioned. The model of the estimation process for planning, scheduling and controlling object-oriented, component-based and web-based software development projects is demonstrated.

1 INTRODUCTION

The development of a large software system is a time and resource consuming activity. Even with the increasing automation of software development activities, resources are still scarce. Therefore, we need to be able to provide accurate information and guidelines to managers to help them make decisions, plan and schedule activities, and allocate resources for the different software activities that take place during software evolution. Software metrics are thus necessary to identify where the resource issues are needed; they are a crucial source of information for decision-making.

The EN ISO 9000:2000 [2] suggests that all quality management processes should use statistical techniques to measure process capability and product characteristics. In [3] the Measure, Analyze and Improve methodology establishing a noble purpose for metrics is introduced. It demands measurement of customer satisfaction in quantitative terms and also requires management to use metrics as an important input for planning, monitoring, and managing the project, and also for controlling the quality of the product.

In the last decade, many companies have started to introduce Object-Oriented (OO) technology into their software development environments. OO analysis/design methods, OO languages, and OO development environments are currently popular worldwide in both small and large software organizations. The insertion of OO technology in the software industry, however, has created new challenges for companies, which use product metrics as a tool for monitoring, controlling and improving the way they develop and maintain software. Therefore, metrics, which reflect the specificities of the OO paradigm must be defined and validated in order to be use in industry. Some studies have concluded that “traditional” product metrics are not sufficient for characterizing, assessing and predicting the quality of OO software systems.

2 MODELLING METRICS SYSTEM

A software development project is a complex process that requires careful planning and scheduling. In addition to the software construction process there are generally other related activities which are preliminary, supporting, or by-products of the software development effort. A successful software product delivery demands appropriate attention is paid to creative content and production activities as well as the software construction.

2.1 APPLICATION MODEL

Measurement and Analysis supports all process areas by providing practices that guide projects and organizations in aligning measurement needs objectives with a measurement approach that will provide objective results that can be used in making informed decision and taking appropriate corrective actions. The metrics application framework is focused on four management areas illustrated in Fig. 1.

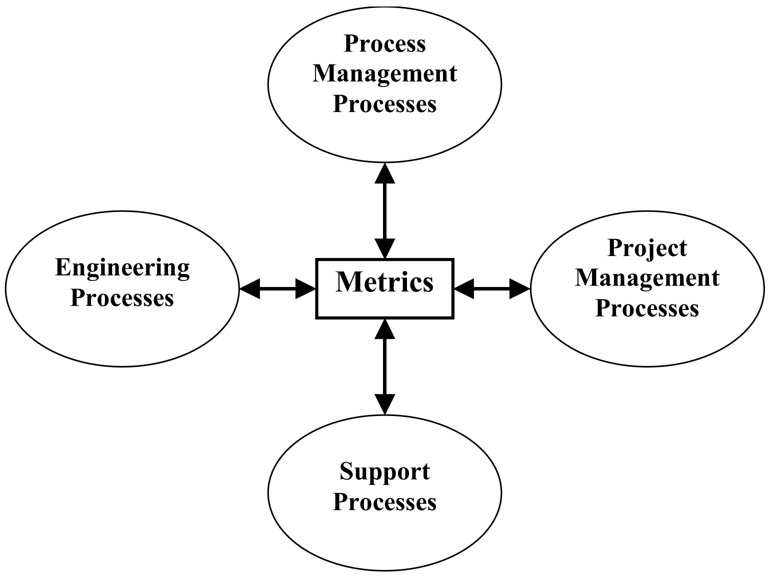

Fig. 1: *Metrics application model*

Quantitative management of all the four areas is the objective of application. Metrics will help the manager see the management details in a structure manner, and move from somehow getting the results to optimizing resource utilization as well as performance. What was an unbroken black box, taking inputs and delivering outputs, now will be broken into micro projects within the projects. Metrics exposes the hidden processes and brings in visibility to the degree and extent one desires and plans for.

In self-organizing systems such as a committed human system the moment of observation is also the moment of action. In such a highly pitched environment metrics and the very act of creating metrics bring about process innovation.

2.2 APPLICATION STRUCTURE

When metrics penetrate into management areas, one can see an application structure evolving in each management area. The application structure consists of identified sub-areas and the associated metrics taxonomy.

The complexity of the application structure increases when the associated metrics taxonomy sizes increase. After a few cycles, the application tree reaches and settles down at a critical size until the next phase change precipitates.

2.3 METRICS SYSTEM

Metrics are best viewed as a system. We cannot design metrics in isolation from environment. Metrics are connected to measurements by mapping rules. Metrics are connected to goals through decision rules. The architecture of metrics system is built around the information highway of organizations, which feeds decision centres. The objective of such a metrics system is to provide model-based decision support.

Designing metrics system architecture is the first step in metrics planning. The second step is to implement the system by working out a set of phased operational plans. Managing these two steps – design and implementation – is what metrics application is all about.

The architecture treats the organization as a network of processes delivering results to the customer. The concept of a value chain is used to identify processes for measurement. Processes that do not add value are not considered.

Models – Knowledge Capsules. Models are abstractions of realities, which allow us to learn by inquiry as an economic alternative to trials. Models help in visualizing the process behaviour. Models can be in the form mathematical equations, graphs and matrices that would represent real-world entities. They also help in forecasting, prediction and what-if analysis.

Metrics – Indicators, Signals. That metrics support the subsequent information processes in the architecture is best seen from a communications system analogy. Well-designed metrics can play the role of corporate signal generators, feeding information networks in the organization. The signal must emerge above noise and communicate messages clearly.

2.4 MODEL OF ESTIMATING PROCESS

When measurements embrace a structure or system, they become more meaningful indicators called metrics. The structure could be a simple algebraic formulation or model.

Moving from measurements to metrics is like moving from observation to understanding. Metrics are conceived by the user and designed to reveal a chosen characteristic in a reliable and meaningful manner. Then these metrics are mapped to ongoing measurements, to arrive at a best fit. The rules for mapping metrics to measurements depend on the problem one wants to address. The mapping rules could be tentative, and the metrics choice could be heuristics.

Thinking of metrics for project management makes one consider the project management structure that one is currently using or that one wants to use. Model of estimating process in software development is illustrated in Fig. 2.

There are set of project factors listed below, which provide the necessary focus on each separate aspect of a project, while maintaining sufficient context to measure and assess the project as a whole:

Scope - identification of software elements to be designed and constructed.

Qualifiers - measure of size, complexity and reuse.

Technologies - impact of environments, techniques and tools.

Software Effort
Activities - relative proportion of effort in various aspects of development.
Metrics - parameters associated with productivity.

Project Effort
Production - Jobs non-software engineering activities.
Bugs - defects discovered in the product which require to be fixed.

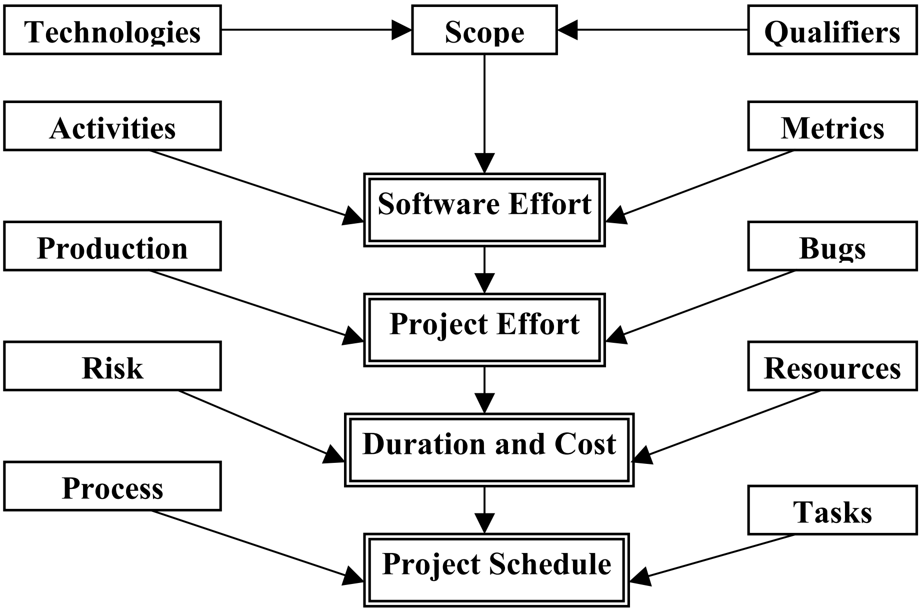

Fig. 2: *Model of estimating process*

Duration and Cost
Risk - identification of adverse factors and their potential impact on development.
Resources - project staffing, skill levels and organisational structure.

Project Schedule
Process - sequential or parallel activity defined by phase, iteration and task structure.
Tasks - allocation of scope and jobs as a work package for specific resources.

PROJECT SCOPE AND BASE PRODUCTIVITY METRICS

The number and type of each scope element provides quantitative data that is key to determining the resulting project estimates. A simple but effective first-cut estimate can be obtained by summing the number of scope elements of each classification and multiplying by the related productivity metric.

SCOPE CLASSIFICATIONS

In the model is defined a number of scope classifications that reflect standard UML scope types. These are the scope elements that are of most interest from an estimation perspective.

A scope classification defines three metrics: concept, discovery and concrete.

Concept Metric - represents a prediction of the effort to analyse, design and build scope elements from a high-level analysis specification.
Concrete Metric - represents the construction effort to design and build from a low-level design specification. Clearly concept metrics are larger than concrete metrics to reflect the earlier stage in the life-cycle, and the likely scope population increase during analysis and design.
Discovery Metric - represents the effort required to replace a high-level analysis specification with a low-level design specification.

Each concept can be defined in terms of a number of concrete elements that implement it and a discovery effort involved in identifying them. Therefore a concept metric can be calculated from one discovery metric plus a number of concrete metrics based on the population ratio.

Each scope type defines a set of metric values that can be applied to calculate effort for elements of that scope type, as follows:

- a base productivity metric and activity profile for concept, discovery and concrete elements.
- baseline adjustment metrics.
- a population ratio that predicts how many concrete elements is required to implement each concept.

3 CONCLUSIONS

Metrics are seen as force multipliers in improvement initiatives and quality movements. On the one hand, the ability to improve is aided by the ability to measure. By integrating knowledge and providing better communication, the resources are better utilized and efforts are better rewarded. Structured thinking a prerequisite for metrics has paved the way for system creation in unexpected areas. For instance, inspired by metrics data patterns, estimation models for bug fixing have been constructed and as a sequel the bug estimation task has been refined and redefined in many organizations.
Progress in metrics data analysis has created new and economic ways of creating knowledge assets in organizations. In that sense, metrics is a rudimentary knowledge engine. Constant interpretations of metrics inject a stream of values into the organization, some temporary, many more enduring. The learning process being what it is, experimental values and tentative knowledge structures all became part of the global repository of knowledge assets.

ACKNOWLEDGEMENTS

This work has been done within the project CEZ:J22/98:262200012 “Research in Information and Control Systems.

REFERENCES

[1] Pandian, C., R.: Software Metrics, A Guide to Planning, Analysis, and Application, Aauerbach Publication, USA, 2003, ISBN 0-8493-1661-8.

[2] EN ISO 9000:2000, Quality Management Systems – Fundamental and vocabulary, European Standard.

[3] EN ISO 9001:2000, Quality Management Systems – Requirements, European Standard.