V7: Measurement

Measurement is potentially one of the most important tools for software engineering management, quality management, and process improvement. The ability of numerical data to focus on the essence of a process or product and provide unambiguous data that can be assessed, analysed, communicated and reasoned about as the basis for decision making is not matched by any other form of information.

What it is for:	To inform decision making with useful graphics or analyses.
When to introduce:	When decision making is not supported by quantitative information
When to use:	Measurement is applicable to most software engineering activity.
When not to use	When there is no reason to measure; do not begin collecting measurement data when there is no clear and explicit reason to collect data.

Ironically the importance of measurement mitigates against its take up. There is a large and intimidating body of literature; reports, papers, books, and tools for measurement. It is frequently adopted as a major component of process improvement. ‘Measurement programmes’ are initiated in organizations to establish a software measurement capability. This gives the impression that introducing measurement to working practice is major investment. If it is introduced as a major initiative or administrative exercise, and perceived as intrinsically complex, the probability of success is low.

Measurement is far more likely to be introduced successfully and be genuinely useful if it is introduced locally and incrementally. The process for introducing measurement is:

1. Establish a specific need for specific measurement data[1]. This may be to (characterize, evaluate, predict, motivate, etc.) the (process, product, model, etc.) in order to (understand, assess, manage, engineer, learn, improve etc.)[2] it. For example to characterize a development process in order to improve it. It is important that the need is specific in order that meaningful questions can be asked about he need.

[1] This first step is the Goal of Vic Basili’s Goal, Question, Metric technique.

[2] The TAME Project, Victor Basili, IEEE Transactions on software Engineering June 1988.:

2. Analyse the need by asking questions. These questions should refine the goals into a set of quantifiable questions that will specify the metrics. The questions explore the understanding of the products or processes to be measured. For example, the need to improve a development process may provoke the question ‘what is improvement?’, leading to a refined goal of reducing rework, leading in turn to the question how is rework defined? This could lead to a definition of rework as fixing defects.

(NB These two first steps need not be performed as ‘stand alone’ measurement activity. In many instances the goals and questions may have already been dealt with in other work to understand and improve aspects of development practices or software products. In such cases a review of the level of understanding, and a brief reiteration and confirmation of goals and questions may be sufficient. For well understood processes and products the development of supporting metrics is natural and almost inevitable.)

3. Define the Measures. When suitable exploratory questioning has been performed the metrics will be straightforward to identify. The term ‘measure’, or ‘metric’ is slightly misleading. Initially the metrics will be items to be counted. In the early stages measurement is a matter of categorizing items and counting them; software measurement is better described as ‘software counting’. The definitions are criteria for determining the category of an item to enable a decision to be made whether it is to be included in a count. For example, an email arrives from a user. Is it an observation, change request, or a defect to be included on the defect log? The definition, or criteria, for a defect will determine which. If it is a defect it can be logged as such and added to the count. Measurement in the broadly recognized sense; continuous measurement of mass, length and time, with varying degrees of precision – 1.0, 1.01, 1.001, etc. is rare in software development. Where it is found it is usually measurement of things not specific to software; time, cost, effort etc.

   When the definitions or criteria have been identified they should be carefully reviewed and made widely available at all times to those involved in either collecting or using the data. Without the definitions the data loses all value.

4. Data Collection: Data is collected, stored and made available to provide the required information. In most cases such data will already be readily available, requiring little in the way of specialized measurement data collection mechanisms. For example, if rework and defects are the focus of the measurement activity a defect log of will probably already be in place, simply as a necessary tool for managing defects. Counts, in particular

cumulative counts, of defects, analysis of defect types , analysis of severities and fix effort should be relatively simple and straightforward. Where a requirement for new data collection mechanisms, and central repositories is suggested the value of these should be reviewed. They are probably not required.

5. Analysis: The analysis of measurement data should remain simple. Minimize arithmetic operations on the data so far as possible, especially multiplication and division. There is a tendency to apply statistical analyses to measurement data – calculating means and standard deviations - that has little value and tends to obscure rather than reveal information. Graphical presentation – scattergrams, cumulative counts and other time series type presentations can be very informative – revealing information in a way that arithmetic analyses rarely do.

6. Verification and Validation: These are necessary activities, but unpopular and rarely performed. This can lead to a loss of value and discrediting of measurement data. Verification is simply checking that the data being collected is representative, of the required accuracy and has no hidden bias. Critical sampling of the data being collected and the data gathering mechanisms will increase confidence in the data. Validation is checking that the types of data and the analyses are providing the types of information required by the original goals. Here the G/Q/M model reveal a further valuable facet by providing a traceability mechanism. By tracing the information back through the measurement definitions and criteria, the questions, back to the goals the adequacy of the measures, their scope and applicability can be assessed and developed as understanding of the products or processes being measured grows.