



Chapter 6

Software and Software Process Measurement

Abstract To investigate whether a (deployed) software process achieves its goals, some form of validation is needed. In many cases, a measurement program can provide a useful basis for such validation, providing information about performance and quality of software and software processes, and thus helping to find gap and identify improvement opportunities or needs. In this chapter, we lay the foundation for such measurement programs. We discuss the measurement basics and provide an overview of different measurement approaches such as the Goal Question Metric (GQM) paradigm. In particular, the measurement of processes and their quality is introduced, using the process quality model Gokyo Ri. Additional information about software processes can be gained from measuring related objects such as the software created or the service provided.

6.1 Introduction

6.1.1 Why measure?

One of the first to argue for the use of metrics in software development was Tom Gilb who compared the software development process to a ship navigating at sea where the ship's captain needs a number of instruments to succeed. Similarly, "the captains of the software ships will be more likely to succeed in meeting their specifications if the instruments for measuring the development process are sufficiently detailed and accurate, and used at a sufficiently early point in the work process" [7, p. 50].

If you can't measure it, you can't manage it. This common saying (alternatively stating "... can't improve it") is variously attributed to a number of famous people such as management guru Peter Drucker or W. Edwards Deming. Nevertheless, it over-simplifies the topic since there are many properties and objects that are important but cannot be measured and still must be managed or improved. On the contrary, there are good reasons to state that "if you don't understand it, you should not mea-

sure it” because otherwise one will measure the easy-to-measure properties rather than the important ones.

Put differently, it is important to analyse what information can really be gained from measurement and measurement data. Some organisations have a tendency to over-interpret measurement data because they provide objective information about the object measured. While this is in general true, it is not always true, and even if the information is objective, it rarely includes *all* relevant information.

Reasons for using measurements. For a variety of reasons, measurements are commonly used in the context of software processes. Two of the main reasons for measuring software processes are:

- monitoring and control of the process: regarding software processes, this may refer to an individual instance of the process such as a specific project. In this case, one usually starts with planning, i.e. setting target values, and then regularly comparing actual values against these targets. In this case, the main purpose of measurement is to provide the information needed to manage the project or, more general, the development effort. On the other hand, the resulting measurement data can then feed into the planning and estimating future work.

Regarding processes in the early stages of the software life cycle, measurements usually provide limited benefit and are therefore rarely used.

At a late stage of the software life cycle, operations and service management are also commonly monitored and measured in order to manage and control the process, for example by measuring the number of open incident tickets and reacting if this number moves outside the target range.

Since operations and service management processes tend to be fairly standardised, defining meaningful measurements for these processes is easier than e.g. for development, and therefore more widespread.

- process improvement: measurements can also help to identify weaknesses in the process, both the process model and the performed process. For example, such weaknesses may consist of process steps that are inefficient or lead to many problems. Once such weaknesses have been identified, they can be addressed and the process improved.

The combined use of measurement data for process improvement and for managing projects is for example described in the Experience Factory concept, see Sect. 5.7, where data are collected to improve processes and to plan and support future work.

6.1.2 Measurement Terminology

Before discussing measurement in more detail, we first introduce the relevant terminology. Two important and closely related terms are measure and measurement: