Chapter 1.16 Software Metrics and Measurements

Michalis Xenos
Hellenic Open University, Greece

INTRODUCTION

In the past few years, a large number of e-government and e-commerce systems have been developed, thus resulting to a constantly increasing number of software developers involved in software development for such systems. To ensure the production of high quality e-government and e-commerce systems, it is important for developers to collect and analyze measurable data that guide estimation, decision making, and assessment. It is common sense that one can control and manage better what he is able to measure.

Although there are major differences between e-commerce and e-government (e.g., access, structure and accountability; Jorgenson & Cable, 2002) there are no significant differences in terms of software metrics that can be applied to both. Metrics are used in e-government and e-commerce software development to measure various factors related to software quality and can be classified as product metrics, process metrics and recourse metrics. *Product metrics* are also called software metrics. These are metrics that are directly related to the product itself, such as code statements, delivered executables, manuals, and strive to measure product quality, or attributes of the product that can be related to product quality. *Process metrics* focus on the process of software development and measure process characteristics, aiming to detect problems or to push forward successful practices. *Resource metrics* are related to the resources required for software development and their performance.

This article focuses on product metrics and on how such metrics can aid in design, prediction and assessment of the final product quality, provide data used for decision making, cost and effort estimation, fault prevention, testing time reduction, and, consequently, aid in producing better software for e-government and e-commerce systems.

BACKGROUND

Measurement is the process by which numbers or symbols are assigned to attributes of entities in the real world so as to describe such entities according to clearly defined rules (Fenton & Pfleeger, 2004). In software development, measurements are conducted by using metrics. A *metric* is an empirical assignment of a value to an entity aiming to describe a specific characteristic of this entity. Measurements have been introduced into the e-government and e-commerce software development process in order to satisfy the need to control software development and produce higher quality results.

Since the mid 1970s when the first software metrics were proposed, a large number of metrics have been proposed in the following years. The proliferation of metrics was followed by more practical proposals on how to interpret results from metrics (see Shepperd & Ince, 1990) and methods combining metrics into measurement methodologies (see Xenos, 2003).

Public or private entities involved in software development for e-gsovernment and e-commerce applications can select from a variety of applied metrics those that are more suitable to be included in the development process (e.g., see Goodman, 2004; Kan, 2003). Therefore, taking into account the volume of literature that exists about software metrics, it is no more a question of finding metrics for an e-government or e-commerce project, rather than selecting the appropriate ones and extensively training engineering teams to utilize them (Hirsh, 2005). Given the large number of metrics (measuring almost everything), any attempt to select a metric without basing the selection on a detailed breakdown of the development needs and an extensive investigation of the proposed metric's applicability would result in minor benefits from its use or no benefits at all. To benefit from the use of metrics, apart from fully understanding the various existing metrics, one should also define well *why* he wants to measure, *what* to measure and *when* is the right time to measure it.

So the first question is: *Why use metrics?* The answer to this question is that metrics are needed to provide understanding of different elements of e-government and e-commerce software projects. Because it is not always clear what causes a project to fail, it is essential to measure and record characteristics of good projects as well as bad ones. Metrics provide indicators for the developed software. As Ragland (1995) stated, indicators are metrics or combinations of metrics that provide insights of the software development process, the software project, or the product itself. Measurements aim at the assessment of the status of the development process and the developed product. Therefore, metrics can be used for performance evaluation, cost estimation as Stamelos and Angelis (2001) have proposed, effort estimation, improving productivity, selecting best practices and—in general—for improving the quality of e-government and e-commerce systems.

This discussion leads to the next question: *What to measure?* As previously mentioned, process and product are what we need to measure. One may argue that, since the result of e-government and e-commerce projects is software, what we need to measure is only software. This is not true. According to Deming (1986), if the product you have developed is erroneous, do not just fix the errors, but also fix the process that allowed the errors into the product. This way you will not have to keep fixing the error in subsequent productions. Therefore, both process and product metrics and measurements are important in e-government and e-commerce software development.

It must be noted that, before selecting the appropriate metrics, it is very important to define the desired product quality characteristics. The selection of quality characteristics aids in defining what needs to be measured and what needs not, depending on the particular needs of the e-government and e-commerce application. In the early 1970s,