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, McCall, Richards, and Walters (1977) defined a framework for measuring such characteristics and proposed the Factors Criteria Metrics model—also known as FCM model—defining what is software quality in terms of subcharacteristics. Incorporating FCM and experience from similar proposals, years later, the ISO standard ISO/IEC 9126 (2001) standardized what product quality is in terms of subcharacteristics. Therefore the definition of product quality is important, as product metrics are used in the software development procedure to measure those product characteristics that are related to product quality.

Having defined the goals and reasons for measuring, the next question is: *When to measure?* Although measurements should be conducted throughout the entire e-government and e-commerce software development life cycle, their scope varies depending on the development phase. Different measurement goals are defined at different development phases, thus resulting into different kinds of metrics. In the early phases of e-government and e-commerce software development, metrics are used mainly for estimation purposes. It is useful to collect metrics relating to different projects as these can serve as historical data for future projects, aiding in better results.

In the intermediate phases of the e-government and e-commerce development process, metrics are used for project monitoring purposes and, in the meantime, code metrics are used to prevent errors. Furthermore, defect reports during testing are used for evaluating product quality and calibrating the measurement methods of the early phases. This purpose is also served by collecting external measurement data following project delivery, namely during the beta testing or maintenance phases of an e-government or e-commerce project. So the time to measure is determined by the requirements and the aims of the measurement program and can vary from a project to another.

Summarizing, using an oversimplifying statement, it could be said that metrics are an important instrument for the development of software to be integrated into e-government and e-commerce systems; metrics aid in making estimations in the early phases of a project, preventing problems in intermediate phases and evaluating quality in the late project phases.

USING METRICS IN SOFTWARE DEVELOPMENT FOR E-GOVERNMENT AND E-COMMERCE SYSTEMS

This section classifies product metrics in two categories—internal and external—provides a short definition and examples of each category, and discusses their advantages and disadvantages. The section concludes by presenting how these metrics can be combined and used in software development for e-government and e-commerce systems.

Product metrics can be categorized (Fenton & Pfleeger, 2004) as internal product metrics and external product metrics. *Internal product metrics* are those used to measure attributes of the product that can be measured directly by examining the product on its own irrespectively of its behavior. *External product metrics* are those used to measure attributes of the product that can be measured only with respect to how the product relates to its environment.

Internal Metrics

Internal metrics can be classified in three categories based on the product attributes they measure. These categories are size, complexity, and data metrics. As far as internal product metrics in general are concerned, it is important to mention that one of their major *advantages* is that they are easy to automate and therefore data collection can be made in an easy, automated, and cost-effective way. Furthermore, the measurement results can also be analyzed in an automated way using statistical techniques and thus conclusions can be drawn rapidly. Tools such as QSUP (Xenos, Thanos, & Christodoulakis, 1996), Emerald (Hudepohl et al., 1996), GQM automation (Lavazza, 2000), and so forth have rendered internal measurements very easy to conduct. The screenshot from the metrics results