In "*Encyclopedia of E-Commerce, E-Government and Mobile Commerce*", Mehdi Khosrow-Pour (Ed.), Idea Group Publishing, ISBN: 1-59140-799-0, pp. 1029-1036, 2006.

# Software Metrics and Measurements

Michalis Xenos

School of Sciences and Technology, Hellenic Open University,
23 Saxtouri Str, Patras, GR 262 22, Greece
xenos@eap.gr
Tel: +30 2610 367405
Fax: +30 2610 367520

## 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 and 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 '70s 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 and Ince, 1990) and methods combining metrics into measurement methodologies, see (Xenos, 2003).

Public or private entities involved in software development for E-Government and E-Commerce applications can select from a variety of applied metrics those that are more suitable to be included in the development process, see for example Kan (2003) and Goodman (2004). 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 (Hirst, 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 rising 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. Since 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) states, 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 70's, 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 sub-characteristics. Incorporating FCM and experience from similar proposals, years later, the ISO standard ISO 9126 (2001) standardized what product quality is in terms of sub-characteristics. 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 that rises 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 while, 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 and 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 behaviour. *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.

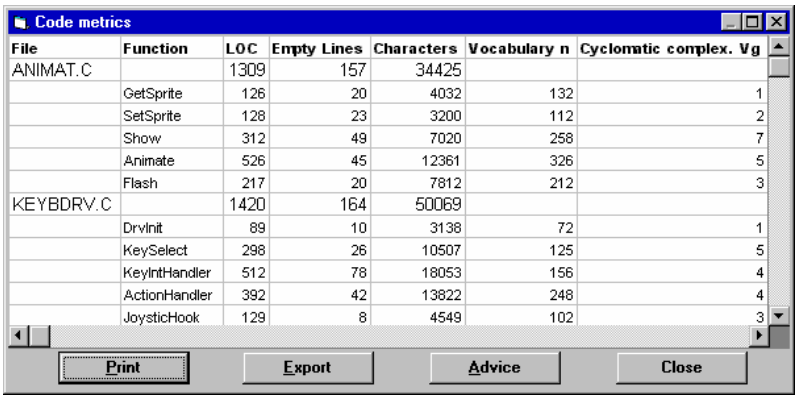


**Figure 1.** The results presentation window from QSUP

## *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 and Christodoulakis, 1996), Emerald (Hudepohl, Aud, Khoshgoftaar, Allen, and Maykand, 1996), GQM automation (Lavazza, 2000), etc. have rendered internal measurements very easy to conduct. The screenshot from

the metrics results of QSUP shown in figure 1 is an example of the simple and automated way in which such measurements can be conducted. For further examples regarding metrics application see (Xenos, 2003).

On the other hand, it should be mentioned that a major among the *disadvantages* of internal product measurements is the fact that they are often difficult to interpret. In other cases, the internal quantities measured are not clearly related to the external quality characteristics that one wants to assess. Such problems can only be solved in the framework of a well-defined measurement method that combines internal and external metrics, as will be discussed hereinafter.

## *External metrics*

Based on the ISO 9126 standard and on similar works such as (Jung et al., 2004), the *external factors* affecting software quality are Functionality, Usability, Efficiency and Reliability. For their definitions see 'Terms and definitions', as defined by (Kitchenham and Pfleeger, 1996). External metrics are used to measure directly these four factors or the characteristics of which these factors are composed. For example, a set of high-level quality characteristics of E-Commerce systems is presented in table 1 (Stefani and Xenos 2001). This is important for the distinction between generic metrics and metrics defined especially for E-Commerce systems.

***Table 1:*** High-level characteristics of e-commerce systems.

| Characteristics of e-commerce systems | Related quality factors |
|---|---|
| Easy access to the web pages of the e-commerce system | Functionality, Usability, Efficiency |
| Easy navigation | Functionality, Usability |
| Adaptation to user profile | Functionality, Usability, Efficiency |
| Search engine service | Functionality, Usability, Reliability |
| Easy exit – undo functions | Functionality |
| Useful help service | Functionality, Usability, Efficiency |
| Electronic shopping cart | Functionality, Usability |
| Electronic shopping list | Functionality, Usability |
| Secure and reliable transactions | Functionality, Reliability |
| Security protocols SET, SSL | Reliability |
| Correct and accurate information about the products | Reliability |
| Direct delivery of the products | Usability, Efficiency |
| Indisputable financial transactions | Reliability |
| Recoverability of products and services | Usability, Functionality |
| Legitimate web site | Reliability |

Unlike internal metrics (measuring software internal characteristics and aiming at relating measurements of such characteristics to these factors), external metrics measure directly these factors or their characteristics. Such metrics can be based on subjective estimates. Among the means employed by external metrics are surveys on user opinion providing valuable measurements for software functionality or usability. Measures like defect reports, or mean time between failures are used to determine product reliability, while measures like memory usage are used to determine efficiency.

As already mentioned, the application of external metrics implies that a certain extent of subjectivity is involved; even metrics that appear to be objective are often characterized by some degree of subjectivity. For example, defect reports seem to be a solid metric that can be used to objectively measure reliability. But the number of defect reports submitted by a user is influenced by issues such as the time and the extent of product usage, the user experience and even the user's motivation to edit and submit a defect report. Therefore, such metrics must be analyzed very carefully and under a framework that will take under consideration such issues.

One of the major *advantages* of external metrics is that they measure directly the desired external product quality characteristics, thus no further analysis or interpretation is needed. Additionally, external metrics contribute to a great extent to what is considered to be one of the main goals of E-Government and E-Commerce quality: user satisfaction. On the other hand, *disadvantages* and problems should be taken seriously under consideration when deciding to use external metrics, the most important of which being that such metrics are not objective and as a result additional effort is required to ensure their objectivity. Furthermore, they are not as cost effective as internal measurements and in many cases it is difficult to conduct measurements due to high error rates especially in cases that error detection techniques have not been used during measurements.

### *Combining internal and external metrics into a measurement method*

As already mentioned, internal and external measurements must be conducted under a well-defined framework with precise goals. Before selecting the appropriate metrics for any project, a *quality manual* should be established collecting and documenting all metrics available for use in the software developing entity. This manual is a basic component of the metrics application process and includes the metrics, the measurement techniques as well as the guidelines for the application of metrics, the data analysis and the corrective actions required for improving the developing process of E-Government and E-Commerce systems. It should also be mentioned that the quality manual includes all metrics that are available regardless of how many times they have been used, or the availability of measurements data from past software development projects.

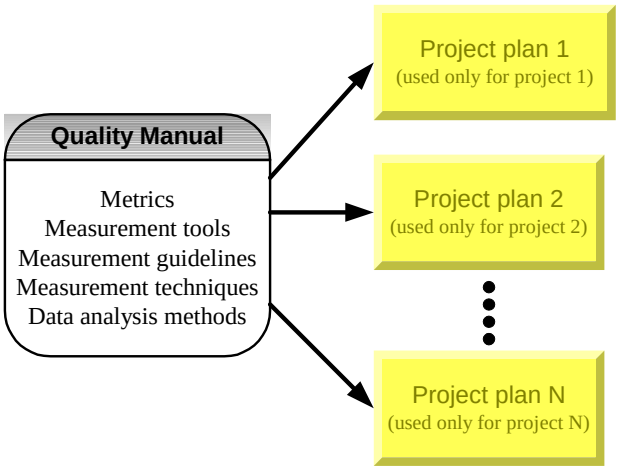


**Figure 2.** Selection of metrics for each project

Then, for each E-Government or E-Commerce project, a set of metrics appropriate for this particular project is selected from the quality manual. The criteria on which the selection of metrics is based are the particular quality factors that the project places emphasis on. This set of metrics is documented –using the guidelines available in the quality manual– and consists the *quality plan* of the particular project. Thus, an E-Government or E-Commerce project quality plan should include all the metrics, measurement guidelines and goals applicable for the project. It is self-evident that the project plan of a specific project may be entirely different from another project's plan and may use a completely different set of metrics. Figure 2 presents an illustration of the above procedure.

The quality plan of each E-Government or E-Commerce project should include internal metrics so as to provide an easy and inexpensive way to detect possible causes for low product quality, as this might be perceived by the end-users, and take early corrective action. It should also include external metrics –applied during alpha or beta testing and post

shipment– so as to measure external quality factors, as well as the soundness of the internal metrics and measurements results or even calibrate internal metrics.

It should be noted that the successful selection of metrics and measurement techniques to be included in an entity's quality manual is heavily dependent on the entity's maturity. The adoption of sophisticated techniques and complex metrics by a company may prove to be ineffective, if it is not supported by years of experience with metrics and measurements and large volumes of data from past project measurements. Software developing companies should always keep this fact in mind and set feasible measurement goals not aiming too high at the early stages of metrics application.

# FUTURE TRENDS

For about three decades now, metrics have been used for the estimation of product related issues (such as product size, required effort, time required for testing, etc.) for early detection and prevention of problems during development and for product assessment after product release. Although in both cases metrics have proved to be successful in practice and have aided significantly towards developing higher quality E-Government and E-Commerce applications, the benefits from the use of metrics are not commonly recognized. This is partly due to the lack of awareness of metrics in small and medium size software developing companies. Although, in large companies metrics are extensively used, in many cases, small and medium size enterprises ones are not even aware of the prospect and benefits of using metrics. However, this is constantly changing. More and more small and medium size E-Government and E-Commerce software developing companies become aware of product metrics and measurements. Besides, the adoption of standards such as the ISO, or assessment in CMM higher levels, has contributed to this change since both standards are encouraging the use of metrics.

Another issue that is expected to change in the near future is the availability of more sophisticated tools. Although many measurement tools are available, using a number of metrics, there are not many tools available yet combining past projects' measurement data with current project data in order to aid in decision making. Combining metrics with decision support techniques, or methods for resolving uncertainty will lead to the development of valuable tools, which can aid towards higher-quality software for E-Government and E-Commerce systems. A recent approach towards this direction (Fenton, Krause and Neil, 2002) is using metrics and Bayesian networks for controlling software development, by automatically predicting defects in the released product.

# CONCLUSION

This article introduced the reader to software metrics that are used to provide insight about different elements of E-Government or E-Commerce systems software. It presented internal metrics that can be applied prior to the release of the product to provide indications relating to quality characteristics, and external metrics applied after product delivery to give information about user perception of product quality.

Software metrics can be used to measure various factors related to software product development. These factors include estimation, early detection and prevention of problems, product assessment, etc. Their utilization within a measurements framework in combination to the use of automated tools can aid towards development process control and higher quality software for E-Government and E-Commerce systems.

Our focus was placed on the particular factors affecting the quality of E-Government or E-Commerce software. Such software can be measured effectively using a combination of generic internal software metrics and external metrics. The former are appropriate for most types of software, while the later are designed especially for E-Government or E-Commerce systems.

# REFERENCES

Deming, W. (1986). *Out of the Crisis*. Cambridge: MIT Center for Advanced Engineering Study.

Fenton, N., Krause, P., & Neil, M. (2002). Software Measurement: Uncertainty and Causal Modeling. *IEEE Software*. 19 (4). 116-122.

Fenton, N.E., & Pfleeger, S.L. (2004). *Software Metrics: A Rigorous & Practical Approach (2nd ed. Revisited Printing).* London: International Thomson Computer Press.

Goodman P. (2004) *Software Metrics: Best Practices for Successful IT Management*, Rothstein Associates Inc.

Hirsh B. (2005). Positive Reinforcement as a Quality Tool. *IEEE Software*. 22(2). 62-63.

Hudepohl, J.P., Aud, S.J., Khoshgoftaar, T.M., Allen, E.B., & Maykand, J. (1996). Emerald: Software Metrics and Models on the Desktop. *IEEE Software*. 13 (5). 56-60.

*ISO/IEC 9126.* (2001). Software Product Evaluation – Quality Characteristics and Guidelines for the User, International Organization for Standardization. Geneva.

Jorgenson D. & Cable S. (2002). Facing the Challenges of E-Government: A Case Study of the City of Corpus Christi, Texas. *SAM Advanced Management Journal* 67(3). 15-21.

Jung H. & Kim S. (2004). Measuring Software Product Quality: A Survey of ISO/IEC 9126. *IEEE Software*. 21(5). 88-92.

Kan S.H. (2003). *Metrics and Models in Software Quality Engineering (2nd Ed)*, Addison-Wesley.

Kitchenham, B., & Pfleeger, S. (1996). Software Quality: The Elusive Target. *IEEE Software*. 13 (1). 12-21.

Lavazza, L. (2000). Providing Automated Support for the GQM Measurement Process. *IEEE Software*. 17 (3). 56-62.

McCall, J.A., Richards, P.K., & Walters, G.F. (1977). *Factors in Software Quality*. (Vols I, II, III). US Rome Air Development Center Reports NTIS AD/A-049. 14-55.

Ragland, B. (1995). Measure, Metric or Indicator: What's the Difference? *Crosstalk*. 8.

Shepperd, M., & Ince, D. (1990). *The Use of Metrics in the Early Detection of Design Errors*. Proceedings of Software Engineering.

Stamelos, I., & Angelis, L. (2001). Managing Uncertainty in Project Portfolio Cost Estimation. *Information & Software Technology*. Elsevier Publications. 43 (13). 759-768.

Stefani, A., & Xenos, M. (2001). *A Model for Assessing the Quality of E-Commerce Systems*. Proceedings of the PC-HCI 2001 Conference on Human Computer Interaction. Patras. 105-109.

Xenos M. (2003). Technical Issues Related to IT Governance Tactics: Product Metrics, Measurements and Process Control. In *Strategies for Information Technology Governance*. Grembergen W. (Ed.), Idea Group Publishing. 216-244.

Xenos, M., Thanos, P., & Christodoulakis, D. (1996). *QSUP: A Supporting Environment for Preserving Quality Standards*. Proceedings of the 6th International Conference on Software Quality. Dundee Scotland. 146-154.

# TERMS AND THEIR DEFINITION

*External Metric* A metric used to measure attributes of the product that can be measured only with respect to how the product relates to its environment.

*Functionality* The external quality factor that refers to a set of functions and specified properties that satisfy stated or implied needs.

*Internal Metric* A metric used to measure attributes of the product that can be measured directly by examining the product on its own irrespectively of its behavior.

*Measurement* A process by which numbers or symbols are assigned to attributes of entities in the real world in such a way as to describe them according to clearly defined rules.

*Metric* An empirical assignment of a value in an entity aiming to describe a specific characteristic of this entity.

*Quality Manual* A manual used by the software developing company that includes the metrics, the measurement techniques, the guidelines for the application of metrics data analysis and the corrective actions required for improving the software developing process.

*Quality Plan* A plan developed particularly for each software project that includes all the metrics, measurement guidelines and goals applicable for this project only.

*Usability* The external quality factor that is defined as a set of attributes that bear on the effort needed for the use and on the individual assessment of such use by a stated or implied set of users.