

## Chapter 3.3

# Enterprise Resource Planning (ERP) Maintenance Metrics for Management

Celeste See-pui Ng  
Yuan-Ze University, R.O.C.

### INTRODUCTION

A typical packaged software lifecycle, from the client-organization perspective, is packaged software selection followed by implementation, installation, training, and maintenance (that includes upgrades). Traditional software maintenance has been acknowledged by many researchers as the longest and most costly phase in the software lifecycle. This fact is no exception in the ERP packaged software maintenance context (Moore, 2005; Whiting, 2006).

According to Ng, Gable, & Chan (2002, pg. 100) ERP maintenance is defined as “post-implementation activities related to the packaged application software undertaken by the client-organization from the time the system goes live

(i.e., successfully implemented and transported to the production environment), until it is retired from an organization’s production system, to keep the system running; adapt to a changed environment in order to operate well; provide helps to the system users in using the system; realize benefits from the system (best business processes, enhanced system integration, cost reduction); and keep the system a supported-version and meet the vendor’s requirements for standard code. These activities include: implementing internal change-requests (initiated by an ERP-using organization’s system users and IT-staff); responding or handling user-support requests (initiated by an ERP-using organization’s system users); upgrading to new versions/releases (introduced by the vendor); and performing patches (support provided by the vendor).”

DOI: 10.4018/978-1-60566-026-4.ch220

In order to achieve the abovementioned maintenance objectives of keeping the ERP system running, adapting the system to a new operating environment, and ensuring the system up to the vendor's requirement for standard code; and realizing benefits such as competitive advantages from the system, the IT department staff has to collect some metrics or relevant data on patches and modifications done to the ERP system so that they can know or can tell the status and the performance of their maintenance activities. The authors in Fenton (1991), Fenton & Pfleeger (1997), and Florac (1992), agree that software maintenance data are useful for planning, assessment, tracking, and predictions on software maintenance. Although, there is a lot of literature on ERP, we find almost no literature on ERP maintenance metrics. Thus, this text is meant to provide some fundamental metrics on ERP patches and modifications which could be useful for ERP maintenance management in order to answer questions on the state of their ERP system, their patch implementation costs, and the ongoing maintenance costs for their previous modification or custom development.

## **BACKGROUND: METRIC**

The *IEEE Standard Glossary of Software Engineering Terminology* (1990) defines a metric as a "quantitative measure of the degree to which a system, component, or process possesses a given attribute." Based on the definition, this text interprets a metric as being derived from data, and as quantifiable, meaningful, and used for strategic, tactical and/or operational purposes. Data (or data item), in turn, is defined as a quantitative indication of the extent, amount, dimension, capacity, size, or characteristic of particular attributes of a task or activity in a process. It can be collected using forms (e.g., change request form, change report, software engineering report), interviews (with the users, testers, programmers, analysts, managers), and via computerized systems (e.g.,

the in-built change management system in ERP, change request database). A goal/question/metric (GQM) paradigm is a systematic way of collecting predefined data, with intended goal(s), and the associated sets of predetermined questions, in order to derive the anticipated measurable metrics. Basili and Weiss (1984) advocate this methodology for collecting valid data. In GQM, which is also known as the top-down approach, the timing (in terms of the software life cycle or activity), interviewees, and reasons for collection are all predetermined.

The literature reports that successful use of measurement/metric program avoids recurring errors (Ebert, Dumke, Bundschuh, & Schmi-etendorf, 2005), improves software maintenance processes at Burrough Corporation (Rombach & Ulery, 1989), and Motorola (Smith, 1993), and improves product quality at AT&T (Fenton & Pfleeger, 1997).

On the flip side, a known metric—together with the context for its interpretation—can determine what data might be collected (Rombach & Ulery, 1989). There are three main purposes of metrics: assessment, prediction, and control. Table 1 shows the main characteristics and objectives of these three metric categories. In addition to this, their application in practice is provided in Table 2.

## **APPLYING METRICS IN THE ERP MAINTENANCE (AND UPGRADE) CONTEXT: AN IMPLICATION FOR MANAGEMENT**

The following discussion focuses on the practical application of the data items or software metrics. As stated earlier, metrics are derived from the data items collected from the activities that produce them. The metrics covered in this section are: (1) the state of the current system relative to the vendor's expectation, (2) the number of modifications or custom developments likely to be affected by each patch implementation, (3)

5 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:

[www.igi-global.com/chapter/enterprise-resource-planning-erp-maintenance/54501](http://www.igi-global.com/chapter/enterprise-resource-planning-erp-maintenance/54501)

## Related Content

---

### Implementation of Web Accessibility Related Laws

Holly Yu (2009). *Encyclopedia of Information Science and Technology, Second Edition* (pp. 1870-1876).

[www.irma-international.org/chapter/implementation-web-accessibility-related-laws/13832](http://www.irma-international.org/chapter/implementation-web-accessibility-related-laws/13832)

### Visualizing IT Enabled Business Process Change

Martijn R. Hoogeweegenand A. T. Kearney (2002). *Advanced Topics in Information Resources Management, Volume 1* (pp. 35-52).

[www.irma-international.org/chapter/visualizing-enabled-business-process-change/4577](http://www.irma-international.org/chapter/visualizing-enabled-business-process-change/4577)

### Financial Trading Systems Using Artificial Neural Networks

Bruce Vanstoneand Gavin Finnie (2009). *Encyclopedia of Information Science and Technology, Second Edition* (pp. 1532-1536).

[www.irma-international.org/chapter/financial-trading-systems-using-artificial/13781](http://www.irma-international.org/chapter/financial-trading-systems-using-artificial/13781)

### Sharing Work Practice in the Distributed Organization

Inge Hermanrud (2012). *Journal of Cases on Information Technology* (pp. 46-60).

[www.irma-international.org/article/sharing-work-practice-distributed-organization/62862](http://www.irma-international.org/article/sharing-work-practice-distributed-organization/62862)

### Enabling Quality Assurance Analytics through the Use of Information Systems: The Case of a Juice Manufacturing Company

Au Voand Rahul Bhaskar (2014). *Journal of Cases on Information Technology* (pp. 1-13).

[www.irma-international.org/article/enabling-quality-assurance-analytics-through-the-use-of-information-systems/109513](http://www.irma-international.org/article/enabling-quality-assurance-analytics-through-the-use-of-information-systems/109513)