

Chapter 4

Application of Software Metrics in ERP Projects

S. Parthasarathy

Thiagarajar College of Engineering, India

ABSTRACT

Business information system is an area of the greatest significance in any business enterprise today. Enterprise Resource Planning (ERP) projects are a growing segment of this vital area. Software engineering metrics are units of measurement used to characterize the software engineering products and processes. The research about the software process has acquired great importance in the last few years due to the growing interest of software companies in the improvement of their quality. Enterprise Resource Planning (ERP) projects are very complex products, and this fact is directly linked to their development and maintenance. One of the major reasons found in the literature for the failure of ERP projects is the poor management of software processes. In this chapter, the authors propose a Software Metrics Plan (SMP) containing different software metrics to manage software processes during ERP implementation. Two hypotheses have been formulated and tested using statistical techniques to validate the SMP. The statistical analysis of the collected data from an ERP project supports the two hypotheses, leading to the conclusion that the software metrics are momentous in ERP projects.

INTRODUCTION

Software process improvement often receives little orderly attention. If it is important enough to do, however, someone must be assigned the responsibility and given the resources to make it happen. Until this is done, it will remain a nice thing to do

someday, but never today. Software engineering process is the total set of software engineering activities needed to transform a user's requirement into software (Humphrey 2005). In other words, software process is a set of software engineering activities necessary to develop and maintain software products. The reason for defining the software process is to improve the way the work is done. By thinking about the process in an orderly way, it is

DOI: 10.4018/978-1-61520-625-4.ch004

possible to anticipate problems and to devise ways to either prevent or to resolve them.

The software processes that are of great concern during the ERP implementation are requirements instability, scheduling and software maintenance (Parthasarathy and Anbazhagan, 2006). Here, we use software metrics to manage and improve these software processes during the ERP implementation. It explains specifically how the software processes can be quantified, plotted, and analyzed so that the performance of ERP software development activities can be predicted, controlled, and guided to achieve both business and technical goals. As mentioned by Parthasarathy & Anbazhagan (2006), though there are a handful of software processes for ERP projects, the processes such as requirements stability, schedule slippage and monitoring the software maintenance tasks are considered more important and account for the performance enhancement of ERP projects. Hence, only these software processes are dealt with using software metrics in the proposed Software Metrics Plan (SMP) developed in this chapter.

Many project managers acknowledge that measurement helps them understand and guide the development of their projects (Fenton et al 2003). They want to select particular metrics, perhaps as part of an overall development plan, but they do not know how to begin. The answers to a manager's metrics questions should appear in the project's metrics plan, so that managers and developers know what to collect, when to collect it and how the data relate to management decisions. The plan enables managers to establish a flexible and comprehensive metrics program as part of a larger process or product improvement program. The research about the software process has acquired great importance in the last few years due to the growing interest of software companies in the improvement of their quality. ERP projects are very complex products, and this fact is directly linked to their development and maintenance.

The total quality management (TQM) notion of prevention rather than correction can be applied successfully in software engineering. The project schedule slippage and tracking problems during maintenance are not uncommon. A key issue in ERP implementation is how to find a match between the ERP system and an organization's business processes by appropriately customizing both the system and the organization (Arthur 1997). This is badly affected due to the instability in the requirements proposed by the customer and the poor capability of the ERP vendor.

The phase "Gap Analysis" in ERP implementation is a step of negotiation between the company's requirements and the functions an ERP package possesses (Carmel et al 1998). Poor requirements specification and its instability badly affects the gap analysis phase of an ERP project which in turn leads to schedule slippage and bubbles of problems during the maintenance phase of the software project. The requirements instability is probably the most important single software process issue for many organizations. The failure of many software projects can be directly linked to the requirements instability (Davenport 1998). Customization, the biggest technology headache is considered as the critical success factor for ERP implementation (Parthasarathy and Anbazhagan, 2007).

Even for those companies that have successfully implemented large-scale information systems projects in the past, ERP implementation still presents a challenge, because it is not simply a large-scale software deployment exercise. Also as ERP implementation is often accompanied by large-scale organizational changes, agile software processes could not create much impact on the ERP projects (Glass 1998). One of the major reasons found in the literature for the failure of ERP projects is the poor software process management. Hence a Software Metrics Plan (SMP) has been developed to deal with the software processes discussed in the literature review and found to be more important for successful ERP implementa-

8 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/application-software-metrics-epr-projects/42249

Related Content

Investigating Impact of Inter-Organizational Factors in Measuring ERP Systems Success: Bruneian Perspectives

Afzaal H. Seyaland Mohd Noah A. Rahman (2017). *Enterprise Information Systems and the Digitalization of Business Functions* (pp. 178-204).

www.irma-international.org/chapter/investigating-impact-of-inter-organizational-factors-in-measuring-erp-systems-success/177343

Critical Success Factors in Enterprise 2.0: The Importance of Business Performance

Ana Silva (2014). *Handbook of Research on Enterprise 2.0: Technological, Social, and Organizational Dimensions* (pp. 244-258).

www.irma-international.org/chapter/critical-success-factors-in-enterprise-20/81109

The Criticality of Transformational Leadership to Advancing United States Government Enterprise Architecture Adoption

William S. Boddie (2009). *Advances in Government Enterprise Architecture* (pp. 130-148).

www.irma-international.org/chapter/criticality-transformational-leadership-advancing-united/4821

Diffusion of Enterprise Resource Planning Systems in Taiwan: Influence Sources and the Y2K Effect

Hsiu-Hua Chang, Chun-Po Yin and Huey-Wen Chou (2008). *International Journal of Enterprise Information Systems* (pp. 34-47).

www.irma-international.org/article/diffusion-enterprise-resource-planning-systems/2134

Microservices Data Mining for Analytics Feedback and Optimization

Kindson Munonye and Péter Martinek (2021). *International Journal of Enterprise Information Systems* (pp. 22-43).

www.irma-international.org/article/microservices-data-mining-for-analytics-feedback-and-optimization/268361