An approach to estimate the size of ERP package using package points

CrossMark

Thangavel Chandrakumar, Sudhaman Parthasarathy *

Department of Computer Applications Thiagarajar College of Engineering, Madurai, India

ABSTRACT

Enterprise Resource Planning (ERP) packages are information systems that automate the business processes of organizations thereby improving their operational efficiency substantially. ERP projects that involve customization are often affected by inaccurate estimation of efforts. Size of the software forms the basis for effort estimation. Methods used for effort estimation either employ function points (FP) or lines of code (LOC) to measure the size of customized ERP packages. Literature review reveals that the existing software size methods which are meant for custom-built software products may not be suitable for COTS products such as customized ERP packages. Hence, the effort estimation using conventional methods for customized ERP packages may not be accurate. This paper proposes a new approach to estimate the size of customized ERP packages using Package Points (PP). The proposed approach was validated with data collected from 14 ERP projects delivered by the same company. A positive correlation was observed between Package Points (PP) and the efforts of these projects. This result indicates the feasibility of our proposed approach as well as the positive climate for its utility by the project managers of future ERP projects. Lastly, we examine the implication of these results for practice and future research scope.

© 2015 Elsevier B.V. All rights reserved.

1. Introduction

Enterprise Resource Planning (ERP) systems are packaged software solutions that automate the business processes of an organization. Analyst firms, such as AMR Research [22,25,29] indicate that there exists a growing demand for such information systems across manufacturing industries and business firms. The high rates of troubled ERP implementations, delayed ERP deployments and outright cancelations call for review of the effort estimation practices to systematically deal with the software size measurement in ERP projects.

ERP systems are information systems that have different development cycles and deployment methods when compared to traditional software products [54]. Conventional wisdom holds that "vanilla implementations" of ERP packages such as SAP R/3 are much more likely to be successful than customized ERP package implementations [11]. However, several previous studies [7,11,48,52,54] have observed that many implementing organizations have had to modify ERP package to meet essential business needs and hence mandate customization during implementation.

Project managers involved in the development of customized ERP packages have to measure their package size in iterations as the requirements would keep changing due to customization. In such ERP projects, estimating the project efforts based on the package size using the existing methods would be a herculean task. Recent studies [13,16,63,64] also indicate that the effort estimation for ERP projects is mostly inaccurate. This

is quite obvious; as software size forms the basis for effort estimation [10]. Hence accurate effort estimation for ERP projects is possible only when software size could be measured accurately in these projects.

Today, it is well known that a typical ERP project includes business process customization and system customization, each of which matches the business needs of the client organization [16]. This in turn implies that the package size in these projects needs to be measured afresh and accordingly the project efforts and costs have to be re-estimated. A potential factor that can affect the accuracies of the established effort estimation models is the size of the software package [67]. For ERP project managers to be adequately prepared for this endeavor, they need to look at alternate approaches for package size measurement in their ERP projects.

Previous research on ERP projects [30,58,59] reveals that the existing methods for estimation of efforts of standard ERP implementation and customized ERP packages are found to be the same. However, these methods which are meant for custom-built software products may not be suitable for COTS products such as customized ERP packages [13, 16,63,64]. Hence, efforts estimated for such customized ERP packages may not be accurate [59]. Examples of underestimated ERP projects [18,43,46] suggest that making average assumptions might have a profound long-term effect on the ERP adopter, meaning a failure to meet growth targets [43], a loss of market edge [18], or even a bankruptcy [46]. Thus, this paper proposes an approach to estimate the size of customized ERP package using Package Points (PP).

Existing methods [35,59] for effort estimation of customized ERP packages either use function points (FPs) or lines of code (LOC) to size the software. The term 'efforts,' also called 'project efforts,' refers to the manpower required for the development of the software product

---

* Corresponding author.
E-mail addresses: t.chandrakumar@gmail.com (T. Chandrakumar), parthatce@gmail.com (S. Parthasarathy).

in a software project [10]. The FP is a unit of measurement to express the amount of business functionality of an information system (as a product) provides to a user. Function points represent software size.

The LOC is a software metric used to measure the size of a computer program. The size is captured by counting the number of lines in the text of the program's source code. The complexity in function point measurement process grows exponentially as the software projects grow in size [39]. Hence, in our approach, we measure the size of customized ERP packages using Package Points instead of the FP and the LOC.

We define the term "Package Points (PP)" as the frame of reference or unit of measure for the size of an ERP package. Package Points are calculated based on the task points and complexity associated with the different ERP package customization approaches namely configuration, functionality, RICE (Reports, Interfaces, Conversions, and Extensions also known as Enhancements) and user interface design [11,19,42].

ERP package size measurement and effort estimation remain critical components in the ERP Requirements Engineering process [16]. Our objective is to provide ERP project managers with a support vehicle which helps them (i) achieve an increased understanding of the customized ERP package size measurement, (ii) estimate effort based on package size for ERP projects, and (iii) gain awareness of how package size and effort estimation in ERP projects matter in terms of efficiency of projects.

Our approach demonstrates that it is possible to find a trade-off between ERP projects that require more effort than expected and those that require less. The practical implications of our approach are that an ERP project manager (i) no longer must live with the current less reliable methods for software size measurement and effort estimation based on these size measures and (ii) can make decisions such as how to accelerate the ERP project in a way that can cause a desired effect on the efficiency of the project while keeping the package size and the efforts intact.

We suggest that our approach should be used in ERP projects that deal with the development of a customized ERP package. We validated this approach with data collected from 14 ERP projects delivered by the same company. A positive correlation was observed between the Package Points (PP) of the customized ERP packages and the actual efforts of these projects. This result indicates the feasibility of the proposed approach and the positive climate for its utility by the project managers of future ERP projects.

The remainder of this paper proceeds as follows. In Section 2 we provide background information on the need for accurate estimation of ERP project efforts based on package size. Also, we present the limitations in the existing methods for customized ERP package size measurement and effort estimation practice for ERP project. We describe our proposed approach in Section 3. Our application of the case study research method is explained in Section 4. It is followed by a brief discussion on the ERP data we used to validate our approach. Section 5 presents a discussion on our results and their implications for practice, followed by validity concerns. Section 6 presents the concluding remarks and areas for future research.

2. Background and related work

This section provides background of size and effort estimation in ERP projects and summarizes related research works emphasizing the need for accurate size estimation of customized ERP packages. Second, it explores current approaches used for ERP package size measurement and the limitations as observed from previous research studies.

2.1. Need for accurate estimation of ERP package size

For large scale IT projects such as ERP, accurate effort prediction becomes a difficult task for the project managers [27,51,61]. Researchers and practitioners acknowledge the need for better software size measurement methods and thereby a better effort prediction system for ERP projects [16,53,67].

In the past decade, a great deal of work has focused on understanding the reasons for inaccurate estimation of project efforts in traditional software projects as well as COTS projects such as the ERP ([13,16,58,59, 64]), the factors that influence the process of effort estimation for ERP projects [1,35,47,49], and the impact of efforts over efficiency of ERP projects [53,61,63] however, till date, no research study has come out with an exclusive method or a model to deal with size estimation of customized ERP package.

ERP packages have been developed by ERP vendors in response to new technologies and emerging business requirements [54]. Although these packages provide competitive benefits to the implementing organizations, Requirements Engineering (RE) based issues (for e.g., effort estimation, ERP package size measurement) in ERP projects are a major concern [12,14,17,62].

About 72% of ERP projects fail to deliver the anticipated benefits [52]. Some of these projects have ended in failure due to poor management of RE processes [50,51,52,55]. These observations from the previous research works are mostly masked as the ERP projects are focused largely from managerial aspects than software engineering perspective. It is observed that ERP package size can be a useful metric for predicting the effort required to develop customized ERP package successfully [30]. Also, Kusters et al. [38] observe the size of an ERP package as a major cost driver in ERP projects.

2.2. Current approaches for ERP package size measurement

The very nature of development of ERP packages differs from traditional software packages [52,61]. Development of ERP packages involves collecting requirements from multiple sources. For example, in the case of a banking application, requirements are collected from the stakeholders of different banks — both private sector and public sector located in different geographical locations. Hence, in packaged software solutions like ERP systems, it is difficult to completely mold the system to fit the existing business processes of the implementing organization. As a result, in many ERP projects, some degree of customization of both ERP systems and business processes is required [54,55].

On the contrary, in traditional software projects, products are designed and built for a particular client organization. These projects use conventional software process models to collect requirements at the initial stage of development, with the anticipation that the resulting product will meet these requirements. Hence the existing approaches to measure the software size in traditional projects are not able to measure the size of the customized packaged software which is being developed in iterations to make it a mature software system.

Most of the existing software size measurement systems have been designed with custom software projects in mind [28,59]. Unlike projects that use a single size measure such as function points or lines of code, typical ERP size measures during the process of customization include counts of the following: users, sites, business units or legal entities, software interfaces, EDI interfaces, data conversion software and data conversions, custom-developed reports, modified screens and ERP modules [51,56,59,60]. Existing methods for ERP package size measurement has its own limitations to handle these customization parameters.

Many effort estimation methods have been proposed during the last two decades. These methods include estimation by experts [34], analogy-based estimation [33] and parametric estimation [36]. From the literature, we find that effort prediction systems based on software size (FPs or LOCs) include parametric models such as Checkpoint [23], Price S/SP [15], SASET [23], SEER-SEM [23], SLIM [23,37], JENSEN [15,44], ESTIMACS [37], BYL [37], WICOMO [37], SYSTEM-3 [37], SPQR/20 [37], object oriented models [31,70] and non-parametric model such as ANGEL [50]. These models have the same limitations as COCOMO 2.0 with respect to the choice of predictor variables where the main predictor variable is either LOC or FPs [16,59].

Very few papers in the literature investigated the size of software applications based on the function point analysis (e.g. IFPUG, COSMIC)