Significance of Software Metrics in ERP Projects

S. Parthasarathy, and N. Anbazhagan

Abstract—Software engineering metrics are units of measurement that are used to characterize the software engineering products and processes. Enterprise Resource Planning (ERP) is a generic term for integrated systems. ERP projects are a subclass of software projects often accompanied by large-scale organizational changes. One of the major reasons found in the literature for the failure of ERP projects is the poor management of software processes. In this paper, we 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.

Index Terms—Enterprise resource planning (ERP), software metrics, software process improvement

I. INTRODUCTION

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 [1]. ERP is a generic term for integrated systems. ERP projects are a subclass of software projects and often accompanied by large-scale organizational changes [2]. 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.

Enterprise Resource Planning (ERP) is primarily an enterprise-wide system, which encompasses corporate mission, objectives, attitudes, beliefs, values, operating style, and people who make the organization. ERP covers the techniques and concepts employed for the integrated management of businesses as a whole, from the viewpoint of the effective use of management resources, to improve the efficiency of an enterprise. Software projects generally exhibit variable returns to scale, and the output from ERP projects is multivariate. ERP projects are a subclass of software projects [2]. Key issues in ERP projects are the larger project duration due to schedule slippage, poor matching between the company’s requirements and the ERP system and less support for maintenance from the ERP vendor [3] [4] [5]. If used properly, software engineering metrics allow us to (i) quantitatively define success and failure, and/or the degree of success or failure of a product; (ii) identify and quantify improvement, lack of improvement, or degradation in our products and processes; (iii) make meaningful and useful managerial and technical decisions [6].

A large body of literature on large scale information system projects like ERP projects has been developed during the past several decades [7] [8] [9]. Apart from the managerial factors, major reasons found in the literature for the failure of ERP projects are the poor management of software processes. This refers to increased project duration due to schedule slippage, mismatch between the company’s requirements and the ERP system, less support provided by the ERP vendor during maintenance. To overcome this failure, we propose a Software Metrics Plan (SMP) containing different software metrics to manage software processes during ERP implementation. These software metrics will help the ERP project team members to keep track of their software processes during ERP implementation.

SMP can be extended further to include other aspects as well. Different projects may require different levels of effort, resources and expertise. Two hypotheses were formulated and tested using the statistical techniques correlation and multiple correlations [14] [15] to validate SMP. The statistical analysis on the data from an ERP project supports the two hypotheses of the study, leading to the conclusion that the software metrics are momentous in ERP projects.

II. PROBLEM DEFINITION

The total quality management (TQM) notion of prevention rather than correction can be applied successfully in software engineering. As discussed earlier, ERP software process improvement requires special consideration and the crucial factors namely requirements stability, schedule slippage and tracking problems during maintenance are to be measured using software metrics in the proposed SMP. Project schedule slippage and tracking problems during maintenance are not
uncommon. A key issue in Enterprise Resource Planning (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 [3]. This is badly affected due to the instability in the requirements proposed by the customer and the poor capability of the ERP vendor (ERP development team).

The phase “Gap Analysis” in ERP implementation is the step of negotiation between the company’s requirements and the functions an ERP package possesses [10]. Poor requirements specification and its instability badly affect 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 failure of many software projects can be directly linked to the poor requirements specification [11]. 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 [16] [17]. Figure 1 shows the gap analysis phase in ERP implementation. Our requirement is that the ERP package must exactly meet the company’s requirements to ensure successful implementation.

III. SOFTWARE METRICS PLAN (SMP)

Software processes have an important influence on the quality of the final software product, and for this reason companies are becoming more and more concerned about software process improvement. The successful management of the software process is necessary in order to satisfy the final quality, cost and time of the marketing of the software products [12]. To improve the software processes [13], a great variety of initiatives have arisen like CMM, CMMI etc. All these initiatives focus on software processes for developing, implementing or improving a quality management system. ERP projects involve software processes that will exist even after the implementation of the ERP system. In fact, the actual processes start only when the going alive is fixed during implementation.

Traditional software projects have requirements collected from the customers for which the software is developed. But in ERP projects, the software is made available to the customers readymade after fine tuning its functionality to match the exact requirements of the customers. The purpose of SMP is to enhance the performance of ERP projects by improving their software processes. The SMP will have a set of well-defined metrics that will deal with measuring the requirements stability, project schedule slippage and tracking the problems arising during the maintenance phase. SMP can also be further extended involving other aspects like effort distribution, productivity etc if desired. SMP will contain the following: (1) Source of data and how it will be captured (2) Periodicity of data capture (3) Formulae (4) Responsibility (5) Measurement process (6) its significance to the project.

<table>
<thead>
<tr>
<th>Name of the Metric</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>Schedule Slippage (SS)</td>
<td>( \frac{(Actual \text{ No. of days} - Estimated \text{ No. of days})}{Estimated \text{ No. of Days}} \times 100 ) per phase</td>
</tr>
<tr>
<td>Requirements Stability Index (RSI)</td>
<td>( \frac{No. \text{ of requirement changed (added/deleted/modified)}}{Total \text{ No. of Initial requirements}} )</td>
</tr>
<tr>
<td>Arrival Rate of Problems (ARP)</td>
<td>No. of problems reported by the customers per month</td>
</tr>
<tr>
<td>Closure Rate of Problems (CRP)</td>
<td>No. of problems closed per month</td>
</tr>
<tr>
<td>Age of Open Problems (AOP)</td>
<td>Sum of time (days per month) that problems have been open / No. of open problems per month</td>
</tr>
<tr>
<td>Age of Closed Problems (ACP)</td>
<td>Sum of time (days per month) that problems have been closed / No. of closed problems per month</td>
</tr>
</tbody>
</table>

Table I shows the various metrics in the SMP. In Table 1, the actual no. of days is the difference between the date when a particular phase/project got completed and the date when a particular phase/project started. Estimated no. of days is the difference between the planned start date of a particular phase/project and the planned end date of a particular phase/project. Source of data for the SS metric is the project schedule actually prepared. The schedule slippage responsibility lies with the project manager. Analysis of SS could be done during every phase of the software project. SS is used to complete the project on time and to avoid schedule slippage. Source of data for RSI metric is the initial set of requirements specified by the client together with the no. of changes made by him. The project manager is held responsible for higher value of RSI. This must be done during each phase of the project. RSI is used to monitor the magnitude of change in requirements. It gives a picture of the clarity of the requirements in the customer’s mind.

The ARP when compared with the CRP will indicate the amount of work still pending and will help him anticipate the work for the next month. CRP when compared with the ARP will indicate the amount of work still pending and will help him anticipate the amount of work which can be completed the next month. AOP will give the time for which a problem of particular severity remains open and help in setting realistic schedule estimates. ACP will give the time taken to close a problem of a particular severity and help in setting realistic schedule estimates. The responsibility for all these software metrics lies with the project manager.

IV. METHODOLOGY AND DATA ANALYSIS

The goal of this study is to establish the significance of software metrics in ERP projects. We use correlation and multiple correlations [14] [15] to test the two hypotheses.
formulated below to investigate the impact of one metric over another. The relationship between two variables is such that a change in one variable result in a positive or negative change in the other and also greater change in one variable result in a corresponding greater change in the other. This is known as correlation. Positive or negative values of coefficient of correlation ‘R’ between two variables indicate positive or negative correlation. We have used Karl Pearson’s Coefficient of correlation which is defined below. Note that R has no units and is a mere number. If there is some relationship between two variables, their scatter diagram shall have points clustering near about some curve.

\[
R = \frac{1}{N} \left( \frac{\sum (x_i \cdot y_i) - (\bar{x} \cdot \bar{y})}{\sqrt{\frac{1}{N} \sum x_i^2 - (\bar{x})^2} \cdot \sqrt{\frac{1}{N} \sum y_i^2 - (\bar{y})^2}} \right)
\]  

Equation (1) is used to compute the correlation coefficient R. Multiple correlations are used to find the degree of inter-relationship among three or more variables. The objective of using multiple correlations is to find how far the dependent variable is influenced by the independent variables. We denote the multiple correlations between \(x_1\) the dependent variable and \(x_2, x_3, x_4, \ldots, x_n\) independent variables, by \(R_{234, \ldots, n}\). The multiple correlation coefficient of \(x_1\) on \(x_2\) and \(x_3\) is the simple correlation coefficient between the observed value of \(x_1\) and its estimated value \(b_{12,3} x_2 + b_{13, 2} x_3\) denoted by \(\hat{y}_1\). We denote the multiple correlation of \(x_1\) on \(x_2\) by \(R_{123}\). Equation (2) gives the value of \(R_{123}\).

\[
R_{123} = \frac{\text{cov}(x_1, \hat{y}_1)}{\sqrt{\text{var}(x_1)} \cdot \sqrt{\text{var}(\hat{y}_1)}}
\]

We set up the following two hypotheses to focus our study and to facilitate statistical analysis:

Hypothesis 1: The metric RSI influences the metric schedule slippage (SS)

Hypothesis 2: The metrics AOP, ACP determine maintenance efforts (ME) (i.e.) person-hours.

<table>
<thead>
<tr>
<th>TABLE II</th>
</tr>
</thead>
<tbody>
<tr>
<td>RSI</td>
</tr>
<tr>
<td>SS</td>
</tr>
</tbody>
</table>

Table II provides the RSI and SS values computed based on the data from an ERP project A. The coefficient of correlation ‘R’ between RSI and SS is calculated as 0.9406. The value of R, being greater than zero indicates a positive correlation between the two variables RSI and SS. Data obtained from the same project A were used to compute the simple correlations between the metrics ME (\(x_1\)), AOP (\(x_2\)) and AOC (\(x_3\)) as \(R_{12} = 0.863\), \(R_{13} = 0.648\) and \(R_{23} = 0.709\) respectively. The multiple correlation \(R_{123}\) computed using the equation (2) gives 0.864. Since \(R_{123}\) is very large, it follows that the variables \(x_2\) and \(x_3\) have considerable influence on the variable \(x_1\). In other words, regression equation of \(x_1\) on \(x_2\) and \(x_3\) will be excellent. This data analysis gives a great deal of support to our two hypotheses formulated in this study. The closest result observed from the value of R and \(R_{123}\) is the strong relationship between RSI and SS and the influence of AOC and ACP over ME. The most reasonable inference that can be drawn from this study is the significance of software metrics in ERP projects as they act as a tool to effectively monitor the crucial phase of ERP implementation (i.e.) gap analysis.

V. CONCLUSIONS AND FUTURE WORK

The conclusions in this paper are of two kinds: 1) conclusions based on the statistical analysis and 2) conclusions on the usefulness of software metrics in ERP projects. Statistical analysis shows that the software metrics proposed in this paper play a significant role for successful ERP implementation. There is sufficient evidence found from the literature review that the ERP projects are damaged heavily by poor software process management, especially the varying requirements due to organizational changes and customization. These varying requirements lead to schedule slippage.

Also we learn that large maintenance efforts are required for ERP projects as the real software processes get life only after ERP implementation. Regarding the usefulness of the proposed software metrics, we conclude that it is a pragmatic, useful method for improving the software processes in ERP projects, because it enables the ERP team to resolve the important issues such as requirements instability, schedule slippage and efforts required for maintenance. Furthermore, the SMP determines decisions for the ERP team and provides against the possibilities of various defects that might arise during or after the implementation. To summarize, software metrics helps the ERP team to keep track of their project and pilot it in the right path as and when there is a deviation.

A future research study could compare the performance of various ERP projects using this SMP with those not using the SMP and the application of other statistical techniques like multiple linear regressions, etc. If the software metrics are linked with some software quality factors, then the performance of the project can be further improved [18]. Hence generating suitable metrics to consider quality aspects of software processes will strengthen the developed software product. It is also proposed to develop a software tool to execute SMP. In the future, we plan to generate a project database to make the results of SMP from each ERP project publicly available for the ERP team.

 VI. REFERENCES


VII. BIOGRAPHIES

S Parthasarathy, Lecturer in Computer Applications, Department of Computer Science and Engineering, Thiragarar College of Engineering, Madurai, Tamil Nadu, India, is a B.Sc., M.C.A., M.Phil., P.G.D.B.A., P.G.D.P.M. professional with four years of teaching experience. He is currently pursuing Ph.D. in Computer Science in the Anna University, Chennai, India. His current research interests are enterprise resource planning (ERP) and software engineering. He has presented more than ten papers in national and international conferences in these areas.

N Anbazhagan, Senior Lecturer in Mathematics, Thiragarar College of Engineering, Madurai, Tamil Nadu, India, is an M.Sc. (Mathematics), M.Phil., P.G.D.C.A. professional with seven years of teaching and research experience. He was awarded the Ph.D. in Mathematics in 2002. He received the Young Scientist Award from the DST, New Delhi, India, in 2004 and the Career Award for Young Teachers from AICTE, New Delhi, India, in 2005. His current areas of research are stochastic inventory modeling and optimization techniques. He has published more than twenty papers in national and international conferences and journals in these areas.