Association for Information Systems
AIS Electronic Library (AISeL)

ICIS 1998 Proceedings

International Conference on Information Systems (ICIS)

December 1998

Critical Success Factors for Software Reuse Projects

Marcus Rothenberger
Arizona State University, Tempe

Uday Kulkarni
Arizona State University, Tempe

Kevin Dooley
Arizona State University, Tempe

CRITICAL SUCCESS FACTORS FOR SOFTWARE REUSE PROJECTS

Marcus A. Rothenberger
Uday R. Kulkarni
School of Accountancy and Information Management
Arizona State University, Tempe
U.S.A.

Kevin Dooley
Department of Management
Department of Industrial and Management Systems Engineering
Arizona State University, Tempe
U.S.A.

Abstract

Systematic reuse is becoming an increasingly popular way to improve software development productivity and quality. The implementation of a software reuse methodology requires substantial investments for the company. The factors that contribute to the overall success of reuse for an organization have been examined in prior research. However, even in organizations that are successful in employing reuse, some projects may fail to achieve the targeted amounts of reuse. This suggests that there are other factors beyond the overall organizational factors affecting the success of software reuse in projects. This research explores the factors contributing to reuse success of individual projects in organizations that practice systematic software reuse methodologies. Structured interviews are conducted with software developers to identify the factors and project data is scrutinized to assess the impact of these factors. Reuse success is measured by the reuse percentage achieved. A large scale survey of software development firms will be used to empirically tests the relevance of the identified factors to systematic reuse in general. We believe that an organization that can identify the factors affecting potential software reuse will be able to better target investments in the improvement of reuse methodology and thus influence the software productivity and quality.

Keywords: Software reuse, IS development, IS development strategies.

1. INTRODUCTION

Systematic reuse of previously written code is a way to increase software development productivity as well as the quality of the software (Banker and Kauffman 1991; Basili, Briand and Melo 1996; Chen and Lee 1993). If previously tested components are reused in a new software project, they are more likely to be error free than new components. This reduces the overall failure rate of the software project. Hence, high quality software can be built by incorporating reused components into the software project. Assuming reuse investments to be equal across the projects of one software developer, there is a direct relationship between the benefits of reuse and the reuse percentage achieved in a project. We can use the reuse percentage as a measure of reuse success on the project level. Companies employing reuse methodologies have to engage in three major activities: the reuse, the retrieval,

and the production of reusable software components. Many publications examine what makes companies succeed or fail in their reuse attempts (i.e., Frakes and Fox 1995, 1996; Frakes and Isoda 1994). However, even within a company that meets the criteria of a successful reuse approach, software reuse is not consistently successful in all projects.

The purpose of this research is to explore what factors can lead to success or failure of employing reuse in projects of software developers that have a overall successful reuse methodology in place. We selected such a software developer and one of its clients for this research to gain depth of knowledge as to what factors influence project success. Looking at only one client eliminates noise and allows us to focus on differences between projects rather than differences between clients. Reuse success is measured by reuse percentage achieved. Based on the results of the case study, a proposition is developed that explains the factors that influence reuse success on the project level in the context of the subject company. The study then explores the applicability of those factors to systematic reuse in general.

2. SYSTEMATIC SOFTWARE REUSE

The traditional method of component reuse is the use of a software repository-based environment. Components are retrieved using tools, such as browsers, which match specifications and documentation of the components against the requirements of the project. The wider the subject domain of the project, the more difficult and inefficient the identification of the correct component becomes.

Business software projects deal with subject domains that are rather complex. One approach for dealing with the component retrieval problem in complex subject domains is the use of an enterprise-level model to facilitate software reuse in custom business software development. Reusable components are mapped to a generic business process and data model. The retrieval of components in such an environment is easier than in a component library environment, since the decision of what parts of the data model to use for a project ultimately leads to the decision of what code components to reuse. For a more detailed description of the enterprise-level model approach to reuse see Rothenberger and Hershauer (1998a). The subject company facilitates reuse with an enterprise-level model. The differences in reuse success across projects are measured as the percentage of development effort reused. Industry and research have established a standard to measure the reuse rate in a traditional repository-based reuse context (Poulin 1997). In previous research, a method has been developed to assess this measure for an enterprise-level model driven reuse context (Rothenberger and Hershauer 1998b).

3. RESEARCH DESIGN

Previous research has explored what makes reuse methods succeed or fail (i.e., Frakes and Fox 1995, 1996; Frakes and Isoda 1994). Findings considered companies and methodologies as a whole. Although the factors previously identified as being crucial to a successful reuse method are constant within one company, reuse success across projects varies. Hence, there must be additional factors at the project level, as opposed to the organizational level, that affect reuse. We are conducting this research in two phases. A field study is conducted in phase one to analyze quantitative and qualitative data from five projects within a single firm that employs an enterprise-level model for development with reuse. The company specializes in the development of business process and accounting systems. This phase of the research identifies a set of factors that affect project reuse success in an enterprise-level model setting.

The second phase explores to what extent findings are generalizable to other systematic reuse-driven software development environments. To do so, we will test the proposition developed in phase one by conducting a large scale survey. This phase of the research consists of surveying software development organizations to empirically test the results and explore if the same factors determine reuse success in other companies employing different reuse methodologies. Our hypothesis is that the factors

identified as influencing project reuse success in an enterprise-level model environment are generalizable to software development firms that employ other methods of systematic reuse.

3.1 Field Study

We are conducting an in-depth case study on a single client's projects of the subject company to identify the factors that influence project reuse success. Project success is assessed for five projects by calculating the reuse rates according to the approach developed by Rothenberger and Hershauer (1998b). In this development context, code components are mapped to the data and process model. Hence, the reuse rate can be used as a measure for both code and design reuse. In the case study, we use the reuse rate as a measure for project reuse success. In spite of the fact that the organizational factors were constant, there is a difference in success across the client's projects in terms of reuse percentage achieved.

To explore what factors cause the difference in reuse success across projects, we identify two people for each project who can make assessments as to what the factors could have contributed to the success of the projects. We will conduct interviews with the head of software development to identify the respective project managers or head designers. The collection of the perceptual data will be conducted in two steps:

1. The project managers/head designers will identify candidate factors for project reuse success.

2. The project managers/head designers will make an assessment of the value for each of the candidate factors in the context of their respective project(s).

To obtain the data for step one, we use a synchronous group support system (GSS) to facilitate the collection of relevant factors among the project managers and head analysts. In order to avoid conformance pressure, we want to ensure anonymity; hence, we chose an interactive GSS style. In a GSS context, we are dealing with a medium group size for which process losses are minimized when employing an interactive GSS style. In the idea-generation stage of the meeting, we will collect the candidate factors for project reuse success as proposed by the participants. Voting will not take place on the candidate factors; however, we will eliminate multiple phrases naming the same construct by facilitating agreement on the meaning of each candidate factor. Structured follow-up interviews will be conducted to fill in the details to obtain a complete picture of the cases.

In step 2, the assessments of the values for the factors will be made on a limited scale (two or three point scales) that allows the project managers/head analysts to reach agreement. In case of a conflict on the value of a factor, we will attempt to resolve it (delphi process). If the conflict remains, we will need to chose the average between the values. A limited scale also enables us to recognize patterns when analyzing the data. The participants will be asked to assess the values for the candidate factors with respect to their project(s). To eliminate project performance differences because of the nature of the client, we chose a series of projects the company conducted with the same client. Hence, we need to anchor the terms in the questions relative to the other projects for this client.

For the final analysis, results will be grouped by the reuse rates of the projects. The relationship between the factors and the reuse success of each project will be explored. We are looking for within-group similarities coupled with intergroup differences. This search for cross-case patterns allows us to avoid premature or false conclusions as a result of information-processing biases. Examining the success for each structural level separately will allow us to conclude where to direct additional effort and investments for improvements. Not every factor has an equal effect on the reuse percentage on all structural levels.

By looking at the projects for one client of one software development company, we are able to eliminate the confounding effects of various uncontrollable variables. Factors such as differences in development methodology and differences as to the nature of the client company are held constant. This way we can ensure that the differences in project reuse success are not because of

factors external to this research. However, to ensure the validity of the results at this stage, we ensured that the selected projects were of sufficient size and complexity to contain rich enough data for our purposes. We also ensured sufficient heterogeneity of the projects indicating that they represent a variety of development projects.

3.2 Empirical Testing

The survey questions will be identical to the questions asked in phase one for the assessment of the factors with regard to the project managers'/head designers' projects. The participants will be asked to answer the same set of questions for two projects: one successful and one unsuccessful project in terms of reuse success. We need to avoid biases because of differences in industry requirements and development methodologies. Hence, we need to ensure that the two projects are selected from clients of the same industry and that the same reuse methodology has been employed for both. We will identify the companies for the survey in two ways: (1) participant companies at conferences on software reuse, such as the fifth International Conference on Software Reuse in Victoria, BC, Canada, are good candidate participants; (2) we will ask the subject company of the second manuscript to identify competitors that are employing software reuse. We are aiming at a sample size of 50.

The survey will include questions relating to the measurement of reuse to confirm the assessment for the project selection. The survey will also include questions regarding the reuse methodology used to confirm the appropriateness of the participating company. Since we are interested in the differences of reuse success within the company setting, we will anchor the survey questions relative to other projects of the company.

4. CONTRIBUTION

Identifying the factors that influence project success in a systematic reuse development environment will allow software development firms to better target reuse investments. Previous research has focused on the critical success factors of reuse methodologies on an organizational level. However, sound methodologies may not be able to achieve their full potential because of implementation factors at the project level. Those factors have largely been ignored so far. The outcome of this stage will be a set of propositions forming a theoretical model that attempts to explain why project reuse success varies in an overall successful systematic software reuse setting. Information rich case studies in a one company setting are conducted to develop the propositions. At this stage of the research, the theory applies only for similar development settings—mainly the enterprise-level model-based reuse approach.

Phase two generalizes the findings to be applicable to any systematic software reuse development environment. This part of the research concludes with a generally applicable theory, which has been developed to describe the factors that have to be considered to ensure reuse success in software projects that are developed using systematic reuse. Since this section validates and generalizes the proposition introduced in phase one, the benefits for industry and research as discussed in phase two apply to systematic reuse in general rather than being limited to the enterprise-level model driven reuse context.

References

Banker, R. D., and Kauffman, R. J. "Reuse and Productivity in Integrated Computer-Aided Software Engineering: An Empirical Study," *MIS Quarterly* (15:3), 1991, pp. 374-401.

Basili, V. R.; Briand, L. C.; and Melo, W. L. "How Reuse Influences Productivity in Object-Oriented Systems," *Communications of the ACM* (39:10), 1996, pp. 104-116.

Chen, D.-J., and Lee, P. J. "On the Study of Software Reuse Using Reusable C++ Components," *Journal of Systems Software* (20), 1993, pp. 19-36.

Frakes, W. B., and Fox, C. J. "Sixteen Questions About Software Reuse," *Communications of the ACM* (38:6), 1995, pp. 75-87.

Frakes, W. B., and Fox, C. J. "Quality Improvement Using A Software Reuse Failure Modes Model," *IEEE Transactions on Software Engineering* (22:4), 1996, pp. 274-279.

Frakes, W. B., and Isoda, S. "Success Factors of Systematic Reuse," *IEEE Software*, 1994, pp. 15-19.

Poulin, J. S. *Measuring Software Reuse – Principles, Practices, and Economic Models*, Addison-Wesley Publishing Company, Reading, MA, 1997.

Rothenberger, M. A., and Hershauer, J. C. "Facilitating the Retrieval of Reusable Software Components with an Enterprise-Level Model," in E. Hoadley and I. Benbasat (eds.), *Proceedings of the Association for Information Systems Americas Conference*, Baltimore, MD, August 1998a.

Rothenberger, M. A., and Hershauer, J. C. "A Software Reuse Measure: Monitoring an Enterprise-Level Model Driven Development Process," unpublished paper, School of Accountancy and Information Systems, Arizona State University, Tempe, AZ, 1998b.