View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by AIS Electronic Library (AISeL)

Association for Information Systems
AIS Electronic Library (AISeL)

SAIS 2008 Proceedings | Southern (SAIS)

3-1-2008

# Traditional and Modern Methodologies in ERP Systems Implementation Using Microsoft Dynamics-GP

Anand Jeyaraj
*Wright State University*, anand.jeyaraj@wright.edu

Vikram Sethi
*Wright State University*, vikram.sethi@wright.edu

Follow this and additional works at: http://aisel.aisnet.org/sais2008

Recommended Citation
Jeyaraj, Anand and Sethi, Vikram, "Traditional and Modern Methodologies in ERP Systems Implementation Using Microsoft Dynamics-GP" (2008). *SAIS 2008 Proceedings*. 31.
http://aisel.aisnet.org/sais2008/31

This material is brought to you by the Southern (SAIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in SAIS 2008 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

# TRADITIONAL AND MODERN METHODOLOGIES IN ERP SYSTEMS IMPLEMENTATION USING MICROSOFT DYNAMICS –GP

**Anand Jeyaraj**
Wright State University, Dayton, OH
anand.jeyaraj@wright.edu

**Vikram Sethi**
Wright State University, Dayton, OH
vikram.sethi@wright.edu

## ABSTRACT

Enterprise Resource Planning (ERP) systems implementation is typically viewed as a customization or an integration effort to fit an organization's business processes or existing systems. However, ERP systems implementation may also be considered as a lifecycle from inception to completion, which is likely to benefit from the use of formal systems development methodologies, including traditional methodologies (e.g. Systems Development Life Cycle, and Object Oriented Analysis and Design) and modern methodologies (e.g. Extreme Programming, and SCRUM). This research examines the use of systems development methodologies in ERP implementations and the extent to which such methodologies aid the successful implementation of ERP systems. The findings are based on reports by multiple project teams adopting different methodologies in implementing Microsoft Dynamics –GP -based ERP systems for real-world organizations.

### Keywords

ERP, Implementation, Systems Development, Methodologies.

## INTRODUCTION

Enterprise Resource Planning (ERP) systems have for long promised streamlined, efficient, and effective end-to-end processes for organizations. Several vendors (e.g. SAP, Microsoft, Oracle) offer ERP solutions (e.g. SAP R/3, Dynamics, Peoplesoft) for large as well as small-and-medium -sized enterprises (SMEs) competing in various industries.

Despite the considerable promise of ERP systems, ERP implementation efforts have seen only mixed results. The lack of overwhelming success in implementing ERP systems has been attributed to different reasons including projects being late or over budget, problems during implementation, issues with the configuration of ERP systems, problems with application integration, etc. (e.g. Scott and Kaindl 2000; Robey, Ross, and Boudreau 2002; Shang and Seddon 2007).

However, one potential reason – the use (or the lack of use or inappropriate use) of systems development methodologies – has not been highlighted in prior literature. While an ERP implementation effort is generally perceived as a customization effort (to fit the ERP software to the organization's business processes) or an integration effort (to tie the ERP software to the organization's existing information systems), it necessarily follows a lifecycle from inception to completion and is likely to benefit from the use of formal systems development methodologies.

This research seeks to examine the extent to which the use of formal systems development methodologies may benefit ERP implementations. [This is a research-in-progress work. Preliminary results will be presented at the conference.]

## SYSTEMS DEVELOPMENT METHODOLOGIES

A variety of systems development methodologies have been proposed over the years; it is estimated that there are more than 1000 methodologies (Jayaratna 1994). Methodologies typically describe a series of practices (including activities, actions, tasks, milestones, and products) that may be followed for successful development of information systems. They provide roadmaps for success in what could otherwise be an uncontrolled and chaotic environment for systems development.

Traditional methodologies have generally been categorized as process-oriented approaches (e.g. Systems Development Life Cycle: SDLC), data-oriented approaches (e.g. Joint Application Development: JAD), and object-oriented approaches (e.g. Object Oriented Analysis and Design: OOAD) (Jeyaraj and Sauter 2007; Vessey and Conger 1994). These approaches accentuate the process of analysis and design that can lead to a successful completion of the finished product, but generally do not cater to changes in functionality in one cycle. SDLC recommends systems development to be accomplished using the stages of planning, analysis, design, development, testing, implementation, and maintenance. OOAD argues that the objects of the system be identified first followed by an examination of how those objects interact with each other; the analysis phase

of OOAD ends with the development of the conceptual model while the design phase of OOAD is used to transform the conceptual model into a working system.

Modern methodologies have recently begun to be employed in systems development. These include Extreme Programming (XP), SCRUM, and Feature Driven Development (FDD) (Beck 1999; Palmer and Felsing 2002; Schwaber and Beedle 2001). These approaches emphasize client satisfaction and the delivery of (even incremental) products while not being completely rigid on the systems development processes, and are agile enough to accommodate any changes desired by clients midway through the development process. XP proposes the stages of coding, testing, listening, and designing, all of which are activities that are repeated frequently with an emphasis on the finished product; XP values the "code" to be the major component without which there is no system. SCRUM recommends standup meetings, frequent stakeholder meetings, and frequent intermediate deliverables with an emphasis on addressing user requirements; SCRUM explicitly recognizes that user requirements often change and that systems development cannot really be planned.

Despite the inherent differences between traditional and modern systems development methodologies, both share the principles of customer communication, planning, modeling, construction, delivery, and evaluation (Pressman 2005). However, traditional approaches may favor rigidity of the analysis and design process (be it incremental, cyclical, or evolutionary, with due allowances for the context in which the system is being developed) while modern approaches may favor attention to the finished product (mostly incremental but which can adapt quickly to changes in requirements). Consequently, both traditional and modern systems development methodologies have implications for ERP implementations.

## RESEARCH METHODS

### Questions

This research is initiated with three research questions [RQs]: a) to determine if the use of a systems development methodology provides significant benefits in an ERP implementation [RQ1]; b) to determine if an ERP implementation benefits the most from the use of a traditional methodology or a modern methodology [RQ2]; and c) to determine specific methodologies that are more suited for a successful ERP implementation [RQ3]. The answers to the above questions are determined as follows: for RQ1, a comparison between a project that adopted a formal methodology with a project that followed an ad hoc methodology; for RQ2, a comparison between two projects that employed a traditional and a modern methodology respectively; and for RQ3, a comparison between two projects that used traditional methodologies or between two projects that used modern methodologies.

### Participants

Students registered for a senior-level "systems development and implementation" course at a large mid-western university are solicited to participate in an ERP implementation project for a real-world SME using the Microsoft Dynamics –GP software (described later). Participants are randomly divided into multiple teams (i.e. consultants) in charge of ERP implementations. [The number of teams and the number of participants in each team are dependent on enrolment; it is expected that there will be multiple teams with multiple individuals in each team.]

### Software

Microsoft Dynamics –GP is an ERP suite aimed at SMEs and provides complete and scalable financial and operational functionalities for organizations to streamline their processes and make better decisions (Microsoft Corporation 2007). The capabilities of Microsoft Dynamics –GP include financials (e.g. cash flow management, fixed assets management, general ledger, payables management, and receivables management), inventory and order processing (e.g. bill of materials, inventory control, invoicing, purchase order processing, and sales order processing), manufacturing (e.g. planning, production, and management), human resources and payroll, and project management. The suite also allows for audit trails, electronic signatures, and analytics.

### Methodologies

Different systems development methodologies are employed in this research. Project teams are allowed to choose from traditional methodologies such as SDLC and OOAD as well as modern methodologies such as XP and SCRUM. [It is stipulated that a specific methodology can be used by only one team. For the purposes of comparison, one team is allowed to follow ad hoc systems development without relying on any specific formal methodology.]

### Tasks

Teams are given a date by which their ERP solutions are due and encouraged to adhere to the principles native to their chosen methodologies. They are also expected to maintain activity logs that document the extent to which their activities were consistent with and deviated from the native principles. Finally, teams are expected to submit intermediate and final reports as well as present their development approach and final solution. The intermediate reports include a business process analysis report, an ERP model report, an ERP system goals report, and an ERP solution report. When examined together with the methodology log, the intermediate reports will provide an understanding of the extent to which the methodology aided the progression of the ERP development and implementation effort. The final report contains the completed ERP solution, an explanation of the ERP project, the role of the methodology used for the project, and the activity log.

### Analysis

Findings are based on a textual analysis of the various documents (such as the final report, intermediate reports, activity logs, requirements specifications, ERP solution, etc.) submitted by the project teams. The researchers will also attend the meetings between the project teams and the client and will have an understanding of the client's needs and expectations. [Coding instruments and methods are being developed.]

### Timeline

The "systems development and implementation" course runs from early January through the middle of March. Teams are expected to complete their ERP implementation projects by early March. Preliminary analysis and results will be ready in time for the conference.

## REFERENCES

1. Beck, K. (1999) *Extreme Programming Explained: Embrace Change*, Addison-Wesley, New York, NY.
2. Jayaratna, N. (1994) *Understanding and Evaluating Methodologies, NISAD: A Systematic Framework*. McGraw-Hill, Maidenhead, UK.
3. Jeyaraj, A. and Sauter, V. (2007) An Empirical Investigation of the Effectiveness of Systems Modeling and Verification Tools, *Communications of the ACM*, 50, 6, 62–67.
4. Microsoft Corporation (2007) ERP Software: Microsoft Dynamics GP Product Overview, available at http://www.microsoft.com/dynamics/gp/product/productoverview.mspx, Updated August 16, 2007, Last accessed January 1, 2008.
5. Palmer, S. R. and Felsing, J. M. (2002) *A Practical Guide to Feature-Driven Development*, Prentice-Hall, Upper Saddle River, NJ.
6. Pressman, R. S. (2005) *Software Engineering: A Practitioner's Approach*, McGraw-Hill, Boston, MA.
7. Robey, D., Ross, J. and Boudreau, M. (2002) Learning to Implement Enterprise Systems: An Exploratory Study of the Dialectics of Change, *Journal of Management Information Systems*, 19, 1, 17-46.
8. Schwaber, K. and Beedle, M. (2001) *Agile Software Development with SCRUM*, Prentice-Hall, Upper Saddle River, NJ.
9. Scott, J. E. and Kaindl, L. (2000) Enhancing Functionality in an Enterprise Software Package, *Information & Management*, 37, 111-122.
10. Shang, S. and Seddon, P. B. (2007) Managing Process Deficiencies with Enterprise Systems, *Business Process Management Journal*, 13, 3, 405-416.
11. Vessey, I. and Conger, S. Requirements Specification: Learning Object, Process, and Data Methodologies. *Communications of the ACM*, 37, 5, 102-113.