Reengineering of Data Warehouse of Public Sector

Ashok Kumar, Anil Kumar

Abstract - Data warehouse is a process of evolution and is a silver bullet of any one sector either public sector or private sector. As is universal truth, after specific time period database of these sector degrade its performance due to several reason, therefore reengineering of data warehouse is needed to increases the performance of data warehouse and reduces the risk associated with the modification of application.

Keywords: Data warehouse, data model, process model, legacy system

1. Introduction

Data warehouse [1, 2] is a collection of technologies aimed at enabling the knowledge worker (executive, manager, analyst) to make better and faster decision. In early nineties, Bill Inmon coined the term data warehouse – A data warehouse is subject oriented, integrated, time variant, nonvolatile collection of data in support of management's decisions.

A relational database is designed for query and analysis rather than for transaction processing. It usually contains historical data derived from transaction data and it also contains data that is derived from sources. It separate analysis workload from transaction workload and enables an organization to consolidate data from other sources.

In addition to relational database, a data warehouse environment includes an extraction, transportation, transformation and loading (ETL) solution on line analytical processing (OLAP) engine, client analysis tools and other application that manage the process of gathering data and delivering it to business process.

The data warehouse is designed to help you analyze data e.g. to learn more about your company sales data; you can build a data warehouse that concentrate on sales.

Data warehouse is subject oriented, integrated, time variant and non-volatile collection of data in support of management's decision making process.

Subject oriented: a data warehouse analyze a particular area e.g. sales Integrated: a data warehouse integrates the data from multiple data sources e.g. source A source B may have different ways of identifying a product, but in a data warehouse, there will be only a single way identifying a product. Time-variant: historical data is kept in the data warehouse e.g. one can retrieve data from three months, six months, twelve months or even older data from data warehouse.

Non-volatile: once a data is kept in data warehouse, it will not change. So historical data in data warehouse should never be alter.

2. LITERATURE REVIEW

Enterprise data warehouse development projects are a subset of information technology development projects which are part of the general product development projects. Budget Over runs, unsatisfactory product performance, late delivery or even failure to meet project deadlines are some of the undesired possibilities in a project. Against the common belief that these undesired eventualities are closely related to technical issues, usually they result from other unrelated reasons like a lack of appreciation and understanding of the vision or an uncontrolled risk management process. Effective management of risks in such a way that it is linked to the triple constraints (i.e. Quality, Time, and Money) of product development projects can help in "shifting the odds to the project manager's advantage". This has been reiterated by the supposition by Pinto [3] that risk management has to do with the development of methodologies "to better assess risk" prior to significant commitment to the project. However, it is the consistency throughout the product development life-cycle that remains important for ensuring effective risk management and the realization of minimized possibilities for catastrophic implications on these development projects.

Enterprise data warehouse implementation can add significant value to the enterprise through increased strategic advantage levels if executed properly yet development projects in this area remain very challenging and involve high levels of risk. Adelman et al [4] aver that risks in enterprise data warehouse development projects remains more than that experienced in other projects.

Projects in the information technology field are notorious for poor performance compared to their counterparts in other industries. McBride [5] found that one-third of all money spent on software is used to repair botched projects and that billions are spent each year reworking software projects that do not match/fit requirements. Enterprise data warehouse development projects have shown to follow the same trend with an embarrassingly low success rate across diverse industries resulting in inadequate or no data warehouses for the companies in question. A survey of 1400 logistics professionals has also reflected on this poor performance by uncovering that only 35% companies in that industry have successfully built adequate data warehouses between 2002 and 2003 [6]. This is despite the continued market growth of around 278% for software, hardware and services, between 2001 and 2006 [6]. Although the survey cited only involved professionals in the logistics profession (a subset of the entire data warehousing industry), it serves as a good indication of the discrepancy between proper project risk management maturity and market growth in the enterprise data warehouse industry.

These findings coincide with the findings of a literature review [7] which examined a number of academic articles, sourced from leading journal in this field. This study reflected on the literature base available in this field through a classification of different research studies and scoring of the groupings in terms of the satisfactory levels. Findings for the "formal theory" in the data warehouse industry, a section which includes risk management, resulted in a low scoring for both the degree of precision of measurement, and the degree of realism of context.

Syed Aris et al [8] propose a risk management model for information system projects with the following four phases, namely: analysis and decision to outsource; selection of service provider; contract management; and on-going monitoring

Celar [9] also proposes a framework for risk management in software maintenance with the following steps: identifying risk sources; identifying risk events and impact on objectives; quantitative and qualitative analysis; and compilation of preventative or mitigation plans with alternatives. Although these theories have been adapted for risk management in system development and outsourced IT projects in general, there is still a need to do the same for enterprise data warehouse development projects.

Although risk management is generally given some attention, the contribution and expansion of the body of knowledge and literature for enterprise data warehouse project risk management remains an important ingredient for minimized project failure in this area. Further adaptation of risk management frameworks like the one outlined in the article by, contributed considerably in guiding risk management framework composition for development projects in the enterprise data warehouse environment.

Celar proposes the classification of risk sources into defined groups; impact analysis, and contingency plan composition with alternatives.

This is in line with the PMBOK generic risk management knowledge area [10] , which outlines all processes involved in the project risk management, namely: risk management planning; risk identification; qualitative risk analysis; quantitative risk analysis; risk response planning; and risk monitoring and control.

However, it outlines typical mistakes done by most practitioners in the industry who over-customize the risk management process to an extent where some important

Aspects are omitted. For instance, the framework excludes the risk planning, identification and monitoring steps yet starts off with classification of the sources into defined groups.

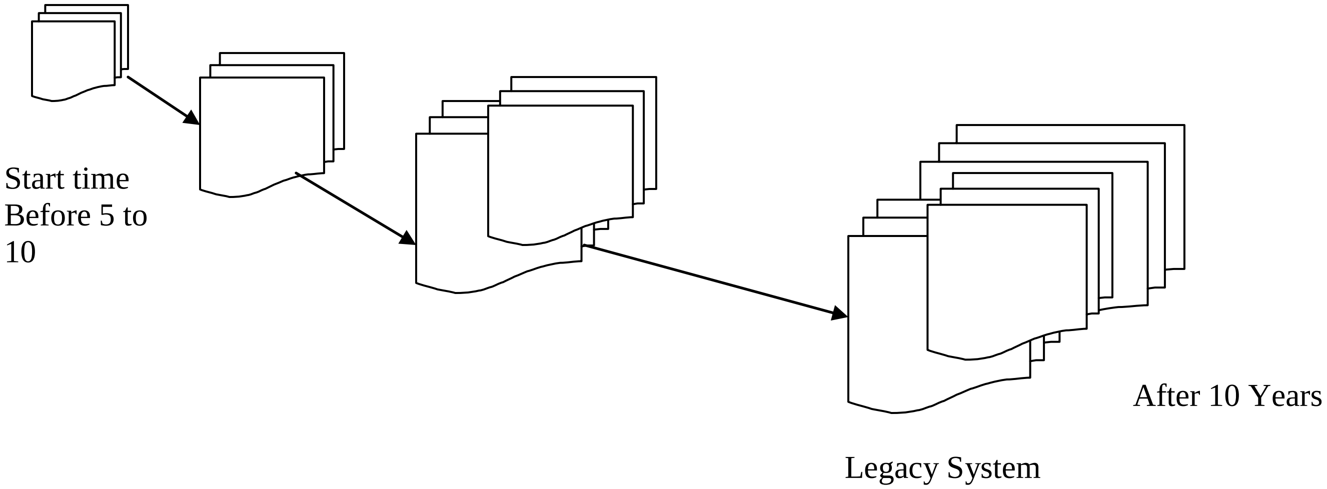
Sometimes these omissions result from risk management process customizations based on assumptions which presume some aspects to be given implicitly although sometimes it result from blatant disregard for proper risk methodologies available in the current literature, a practice which results in improper risk management for the projects in question. Processes and activities included in the PMBOK risk management knowledge area should thus play a guiding role in effective risk management using a suitable risk model for an enterprise data warehouse development projects.

The standard risk model by Smith et al [11] is ideally suited for this purpose due to its ability to emphasis a cause and effect relationship for each risk. This is achieved through the identification of the risk event (cause), relating it to the possible impact (effect) which will result in the total loss, each with an associated probability and drivers.

Having that said: the ability to model project risks in terms of the cause and effect allows for proper appreciation of the total possible loss. This practice can also be seen in different other industries like risk modeling for process plants as illustrated by Visser et.al. [12]. Various theories and models are available in the current literature for risk management in system development and outsourced IT projects, there is still a need to adapt these to cater for specific needs in enterprise data warehouse development projects.

3. Problem Description

As long as legacy system was modest and of manageable size, there was no problem, but as legacy system grew and aged, a number of problems arise, such as new applications became increasingly difficult to implement, especially where there is no desire to overlap with the existing systems. Second difficulty, is that maintenance of existing application systems within legacy systems became more complex and difficult. Third difficulty with the legacy system's growth and aging is need for synchronization of data and processing with the legacy system. An update occurs in one application, and then the update must be reflected in two other applications. This update triggers, yet another activity that needs to occur in yet another part of the legacy system. And so forth. As the time passes growing need of requirement, time to time 'n' number of times in legacy system modification takes place, due to this legacy system grow in volume of data and processing. Then it becomes more un-widely and more intractable.



The problem associated with legacy system, is well known and has been documented. But curing the problem of legacy system has been more difficult then recognizing them.

Therefore after specific time period, maintenance of legacy system's database is very difficult as well as cost of maintenance also increases. Therefore reengineering of data warehouse of legacy system is good choice.

4. Related work

According to William H. Inmon, there are number of approach, which are used to modifying existing system.

The simplest approach is to select an application from the legacy system, where modification is need, remove the application, restructure and rewrite the application along the lines of integrated subject areas and replace the new refurbished application.

This approach is not a good choice, because it is very difficult to pull application out of legacy system, because they are tightly interwoven with other application within the legacy system. In addition modification takes place in the system is not documented due to this further modification create problem for maintenance of modules. However, this is not a good choice because these do not provide feasible solution of existing system.

Second approach is that select a subset of several applications and reshaping those applications according to the corporate plan for integration. Here also we will face difficulties with trying to reshape some part of an application then replacing it. Due to this complexity of application / system increases and high risk factors that are associated with the approach combine to make this option highly unlikely.

Third approach is replacement of whole system and rewrite of all of legacy system all at once. While this approach is used it is associated with very real risks:

Cost of replacement of whole system is very high, sometime it is not desirable to meet the current technology and need of requirement

While whole system is replaced, usually whole database is lost Not support for incremental development

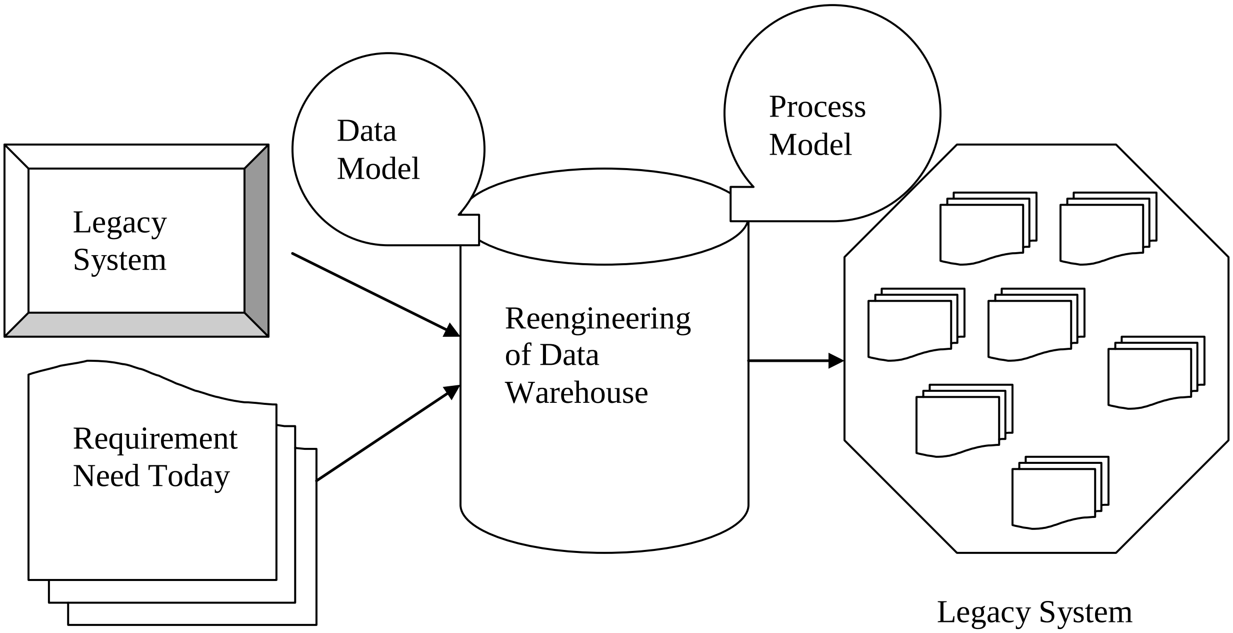
There are several risks associated in changing a complete application all at once.

Once the risk is determined with the system then decision is made whether the system will be change or replaced by new system.

5. Purposed Solution

Here we will introduce a new methodology; Pattern Based Reengineering Methodology that will help us to overcome those entire problems that we will face in above approach for modification of the system. Therefore this methodology helps us in reengineering of legacy system instead of modification of either whole system or a part of system or a single application. While we do reengineering of data warehouse through pattern based reengineering methodology, it will maintain data in database consistent format, addition, and deletion and updating of data from one database will not affect another database and is carried out through integrity constraint and database trigger.

Before reengineering of data warehouse firstly create data model [13] and process models [14] , represents information related to corporate and information processing. Data model [15] is high level abstraction just like top level design.



Data Model: show how information is, or should be stored and used within the corporate sector/public sector. As we know that while data stored in database in consistent format we always use a integrity constraints to store data in database. The specify relationship among data model, which is dynamic link such as user to database, database to data by creating relationship among one database to another database and one user to another. Once you set data model according to requirement to do reengineering of data warehouse there is no need to manage the data model, unless you change the solution design.

Before reengineering of data warehouse first determine on which database or on which data or on which application where reengineering will be taking place. It will determine what is the exact requirement, what will be done, what source of action will be used and what are the changes taking place in application that will not affect corresponding application. Then create static and dynamic links between data items. It will make it easier to traverse the data programmatically from within a policy. During reengineering of data warehouse, data model is used to describe the most important thing in corporate sector/public sector. Purpose of it is to describe the relationship between the data stored about the product and data stored about the organizations that supply the product.

Process Model: during reengineering of data warehouse process model [16] play very important role, which will show flow of information through the system. Each process transforms inputs into outputs. It will determine the way of working and what happens where and when. Same process model is used repeatedly for the development of many applications. It will prescribes, how things must/should/could be done in contrast to the processes itself, which is really what happens.

During reengineering of data warehouse, whenever reengineering applies on a application, then changes takes place in it will be stored in data dictionary, that is used for future references, if anytime, anywhere, any type modification/reengineering is takes place, it will reduces the time to understand whole system and dramatically increases the performance of the system, within low cost and time. Therefore process model play very efficient role during reengineering of data warehouse. It helps us to determine fan-in and fan-out among different application model.

Goal of process model during reengineering of data warehouse are:

- Track what actually happens during a process
- Determine where improvement will be takes place and how efficiently and effectively it will work
- Define desired process and how they should/could/might be performed
- Establish rules, guidelines and behavior pattern which if followed leads to the desired process performance.
- Provide explanation about the rational of process
- It will establish explicit link between processes and requirements that the model needs to fulfill

In pattern based reengineering methodology process model determine the value of fan-in and fan-out, if there is high fan-in that will indicate risk related to the application such as if any of the previous application or task is skipped, then the activity will be delayed whenever next one application depends upon previous one. If there is high fan-out that will indicate risk related to the application such as if single application/task skip then many other application/task may stop working.

The purpose of process model during reengineering of data warehouse is to reduce the risk associated with continuing change of large software that is undergo continuous change or becomes progressively less useful. Reducing complexity in the work done to maintain during reengineering of data warehouse as software system evolve not only today but it also help in future. Whenever risk analysis is determined in early stage during reengineering of data warehouse.

6. Conclusion

Reengineering data warehouse is done in such a way that will determine/reduces risk associated within the corporate/public sector in early phase of reengineering or early phase of software development. And modification takes place in one part of application or 'n' number part of application or in whole application not effect other application that are exist in corporate/public sector. This entire task is performed through Pattern based reengineering methodology.

References

[1] W.H. Inmon, Building the data warehouse. QED. Press/ John wiley, 1992

[2] R. Kimball. The data warehouse toolkit, John Wiley & Sons, 1996

[3] Pinto, K., 2002, Project Management, Research - Technology Management, Industrial Research Institute, Inc.

[4] Adelman, S. Moss, L., 2004, Data Warehouse Risks, Sid Adelman & Associates, Sherman Oaks, CA, http://www.sidadelman.com/3_data_warehouse_risks.htm [Accessed on 04 April 2009]

[5] McBride, S., 2004, Poor project management leads to high failure rate, Available from: http://www.itworld.com/041015poor [Accessed on 10 Sep 2009]

[6] Corporate Executive Board, 2003. Best practices in warehouse management system implementation. Washington: Executive board

[7] Jourdan, Z., Rainer, R., and Marshall, T., 2008. Business Intelligence: An Analysis of the Literature, Information Systems Management, 25(2), pp.121-131

[8] Syed Aris, S., Arshad, N., Mohamed, A., 2008, Risk Management Practices in IT Outsourcing Projects, Universiti Teknologi MARA Selangor, MALAYSIA, 1-3

[9] Celar, S., 2005, Project Management in IT – Projects – A Framework for the risk management approach in SW Maintenance, Annals of DAAAM for 2005 & proceedings of the 16th International DAAAM Symposium, Vienna

[10] Project Management Institute, Inc., 2004, Guide to the Project Management Body of Knowledge - PMBOK Guide, 3rd ed, Pennsylvania

[11] Smith, G & Merritt, M., 2002, Proactive Risk Management: Controlling Uncertainty in Product Development, New York: Productivity Press

[12] Visser, K., Viviers, J., 2008, A risk management methodology for nonMetallic process equipment, South African Journal of Industrial Engineering, Vol 19, Issue 2, Southern African Institute for Industrial Engineering

[13] "Conversion Technology: An Assessment," Data Base,vol. 12&13, no. 411. Surer-Fall, 1981, pp. 39-61.

[14] Ambriola, V., R. Conradi and A. Fuggetta, Assessing process-centered software engineering environments, ACM Trans. Softw. Eng. Methodol. 6, 3, 283-328, 1997.

[15] D.G. Rice and S. Laufer, "Putting Top-Down and Bottom-Up Analysis Together," Database Programming & Design vol. 1. no. 12, December 1988, pp. 46-53.

[16] E. Yu and J. Mylopoulos, "Understanding 'Why' in Software Process Modeling, Analysis, and Design," Proc. 16th Int'l Conj SOBware Engineering, IEEE CS Press, Los Alamitos, Calif., 1994, pp. 159-168.

Dr. Ashok Kumar has received his Ph.D degree from Agra University, Agra, India. He has joined as a Professor in the Department of Computer Science & Application, Kurukshetra University, Kurukshetra – 1361199 (Haryana), India, in June 1982. He has published more than 90 national and international papers. He has attended more than 50 national and international seminars. His area of interests is software engineering, operational research, networking and operating system.

Anil Kumar received his Master degree from IGNOU, India and M.Tech in Computer Science & Eng. From Kurukshetra University, Kurukshetra, India in year 2002 and 2006. He is pursuing Ph.D in Computer Science from the Department of Computer Science & Application – Kurukshetra University, Kurukshetra, India. Currently he is working as an Asst. Professor in Computer Science & Engineering Department in Vaish Engineering College, Rohtak, Haryana, India since September, 2006. He has also worked in software industries for more than three years. His research area includes Software engineering, Reengineering, Software Metrics, Object Oriented analysis and design, Reusability, Reliability.