DATA INTEGRATION: AN EXPERIENCE OF INFORMATION SYSTEM MIGRATION

Inggriani Liem[1], Dindin Wahyudin[2], Alexander Schatten[2]

[1] KK Data & Software Engineering
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
inge@informatika.org

[2] Institute of Software Technology
Vienna University of Technology
{dindin, schatten}@ifs.tuwien.ac.at

Abstract

In this paper we present an experience of migrating a legacy academic information system to a new system which has an entirely different characteristic. To be successful, project managers in such projects will need a comprehensive view of all projects aspects: in particular stakeholder participation and commitment with regards to the software development. In our work, the legacy system was a centralised data processing system having distributed and redundant data representation. The new system supports distributed transactions, and has a single centralised database and supports a superset of responsibilities (data processing and monitoring).

While the new application has been completely redesigned and rewritten, the huge volume of legacy data had to be restructured, integrated and conserved. This was particularly important, since academic data has a long period of validity. The new system has to handle many concurrent users and has strong security requirements. Additionally, a short implementation phase with easy maintainability was demanded resulting in a short learning curve for the end-users.

We will share process and methodology that has been undertaken, and conclude that top level management commitment, business process redefinition, application development & data migration strategy as well as user involvement are the key success factors of new system implementation.

Keywords: Data Integration, Business Process Reengineering, Information System Migration, Stakeholder Management.

1 Introduction

1.1 Legacy Systems

In recent years Institut Teknologi Bandung (ITB) has changed its status from State University (*Perguruan Tinggi Negeri*) to State Owned Legal Entity (*Badan Hukum Milik Negara*). During this transition, a number of transformations have to be established, whereas academic system re-engineering is one of the top priorities. The academic information legacy system was made up of a centralised Java based application. There were three levels of users who produced and utilized these data: Top level (ITB executives), middle level (Faculty), low level (Department and Study Program).

The legacy system operated based on long-used practices; there were neither technical documentation nor standard operating procedures. The application only supported limited business processes for example class registration, student registration, change of study plan, managing exam results, and final grade processing.

Delegated operations which are executed at lower level usually experienced many difficulties by lack of coordination. Centralised processes that are handled by the ITB' centre are easier in comparison. However these centralised processes may cause significant problems, such as issues involved in handling large number of students at a time. This is particularly critical, as these processes are scheduled and must be finished in one week, in which all students have to be registered, student cards must be issued and study plans organised.

Although the legacy system only contains limited processes, the data is large and must be handled permanently. Data duplications occurred since processing must be undertaken in more than one level:

- Data from registration process in central were send to lower levels (faculty and department) for operational purposes, i.e., generate class's presence list;
- Final grades forms were made in central then replicated and distributed to lower level for being processed and then collected again for scanning in central;
- Higher level updates were not possible due to undefined procedures and would see a lot of technical obstacles.

These issues are causing important decisions such as graduation and study termination more difficult, as the data was replicated in three levels of the organisation. The consolidation process consumed a lot of time and forced all levels to have to search through their archives.

The new system would enable users to manage integrated data, and maintain data accuracy and consistency across all operational levels. All contributors (data sources) should be proficient in doing their tasks and follow the integration regulation of the system.

1.2 New System

The new academic information system (called SIX) was designed based on ITB's business processes in accordance to the existing infrastructure condition. SIX acts as an independent application that functions as interface to other systems in ITB. E.g., SIX interacts with the finance system: in this case, SIX only handles the payment status. The rest of financial processes, such as interacting with banks are handled by the finance system.

Based on best-practices, a set of *Standard Operating Procedures* (SOP) was written prior to development. This stage is crucial to develop a foundation to the new application. The SOP documented all business processes to administer all transactions during a student's "life cycle" at university. SIX categorises its users into several groups: *students*, *lecturers*, *administration staff*, *middle level management* (faculty, study program), and *top level executives* in ITB.

The software was designed in three layers:

1. Core Application: accessible only from ITB network by administration staff and executives of ITB. This application handles transactions of main business processes of the system.
2. Internal web based system: accessible only from ITB network, and accessed by trusted user for certain personal transactions, such as checking student's grades.

3. Public web based system: this system is publicly accessible via Internet, and covers public transactions and provides information about academic issues in ITB.

The database was developed based on ITB's business processes - and it employs a hierarchy of transactions, with the academic data transaction set as the top priority.

The difference between the old legacy and the new system could be summarized as shown in Table 1.

Table 1: Comparison between the old legacy system and the new system

Characteristics	**Legacy System**	**New System**
Stakeholder and data ownership	Registrar Office. Faculty (with self initiatives). Department (mostly paper archive).	4 levels: ITB executives –Directorate of Education – Faculty – Department (executives, administration)
Digital data	Distributed, redundant.	Centralised, single data base.
Process	Centralised, using replicated data.	On line transaction, web based system and client server system.
Business process covered by application	A small part of academic life cycle: class and student registration, final grade recording.	Complete academic life cycle: Curriculum, student plan, new intakes management, class registration, registration, final grade recording, graduation, termination status (Drop out, withdrawal), transcript, diploma, course evaluation system
System Scope	Restricted to data processing (EDP).	Data processing, monitoring, decision support system, EIS.
Business process and standard operating procedures	Undocumented.	Well documented, under configuration management control.
Application Software	Not standardised.	Single application for all level of user. Open Source Software.
Development tools	Java platform and MySQL database, Clipper, Office application (at low level user)	Java platform (client server application), PHP, MySQL database (web based application).
Conventional Archives (papers)	Many copies distributed and maintained by each level.	Formal archives must be printed from the database. It ensures integrity and accuracy.

1.3 Related Work

1.3.1 Legacy Information System Migration

Bisbal et al., ([5], [6]) note that most legacy information systems run on obsolete, slow and expensive to maintain hardware and software. Hence traceability of faults is difficult and costly. Lack of documentation and weak coordination and understanding of the internals of the system also contribute to the difficulty. Efforts to integration and expansion are also hard, due to the restricted design of the system. Consequently, during its life-time, a system is subjected to many maintenance activities, which may cause degradation in quality. Such legacy systems are subject to reengineering; particularly, because once the degradation exceeds the system's critical threshold, it would become a significant impediment for an organisation in achieving optimal value from its information system investment. Thus legacy system reengineering and migration has always been a major issue in research and development.

Recent research introduces several reengineering and migration methodologies as well as success stories in specific environments. Ganti and Brayman [11] propose some general guidelines for migrating legacy systems to a distributed environment by examining the business and then re-engineer business processes found as required. It is anticipated that the legacy system migration would cause minimum disruption to the current business environment, Brodie and Stonebraker [9] outline a set of steps for a generic migration strategy. In their method the legacy and target information systems are operated in parallel throughout the migration, which appears to be a very popular approach in system migration. The process itself starts by migrating a small portion of the system such as a database. Then its size is expanded until it covers all the functionality of the legacy system which can eventually be replaced. During this migration process, the legacy and target information systems interoperate to form the operational mission-critical information system.

Wu et al., [13] propose the *butterfly methodology*. This methodology provides guidance for migration of a mission-critical legacy system to a target system. During the migration, the methodology eliminates the need for system users to simultaneously access both the legacy and target systems and therefore, maintaining consistency between these two information systems.

Other researchers such as Bergey et al. [2], provide a comprehensive guideline called DoD guidelines based on actual legacy system migration efforts and evolution of legacy systems from a technical and management point of view. Aversano et al. [1], report a success story in internet migration while salvaging the old centralised mainframe-oriented system by migrating core legacy applications towards Web-enabled client-server architectures.

Bianchi et al. [3] suggest an iterative legacy system migration. The idea is, to preserve the asset of the legacy system, the operational pattern of system's maintainers and users, and sustain the current operation execution continuity during the reengineering process. Bianchi proposes a reengineering process model, where each program needs to be reengineered within a short period of time. This model is applied to an in-use legacy system to confirm that the process satisfies previous requirements and to measure its effectiveness; ensuring that the reengineered system will replace the legacy system to the satisfaction of all stakeholders.

Our approach proposed in this paper, is developed on some of the above mentioned methods; yet we did not employ a particular method, due the complex specification of our legacy system and its implementation environment. Our approach proposes a gradual system migration with emphasize on stakeholder value and their satisfaction.

1.3.2 Stakeholder Management and Value Based Approach

In our approach we take our project stakeholders strongly in our consideration. Stakeholder proposition is described later in this paper. Freeman [10] first proposed in 1984 his stakeholder approach to strategic management, which stated that corporations must consider the needs and demands not only of their shareholders but also those of a wide range of other external constituencies, or "stakeholders." As he implies that the stakeholder management is a prominent requirement in what he called "Managing in Turbulent Times".

Current research in software engineering, which adopts principles in stakeholder approach, is *value based software engineering* (VBSE). Addressing value considerations beyond cost models in the software development context was introduced by Barry Boehm in 1981 [7]. Later on, Boehm proposed a formal value based software engineering agenda [8] that captured the expanding scopes on this burgeoning field. Biffl et al [4] promote the importance of value consideration in software development, which can leverage the quality of the processes.

In this paper, we deal with set of stakeholders with their own varying requirements and expected values. We found that if we carefully managed those stakeholders -also by encouraging their participations and commitment -it has significant impact to the success of this project.

2 Issues

The migration project leads to three main issues:

(a) How to reengineer business process in a changing environment in order to meet new stakeholders' viewpoint.
(b) How to migrate a big volume of legacy data inline with incremental business process formulation, in order to have integrated data in the new system while new transactions must be handled in parallel with the running system.
(c) How to involve and educate end users who have no previous experience on electronic transaction with the shortest training period.

3 Solutions

3.1 Stakeholders Roles in System Migration and Development

As we described earlier, one of our goals in this project was to develop a sustainable system that can support the business processes in our organisation. We employ a good stakeholder management that provides mutually satisfactory relationship among the stakeholders as a critical success factor for the project and the corporate performance- as suggested by Preston et al. [12].

3.1.1 Top Level Management: Software Owner

The academic system was developing consistently with the credo of the Rector: "*Listen, Learn and Change*". Top-level management commitment is the prime factor for changing an academic system to a new one. Development progress was monitored in executive weekly meetings through a visual monitoring system.

The overall system operation and maintenance was under the responsibility of directorate of education, which is accountable for the registrar office function.

3.1.2 Developer

The development team consists of (a) Technical assistant of Vice Rector for Academic Affairs who plays the role as the requirement engineer and quality controller of the project (b) four part time functional analysts and network experts (c) one database specialist who acts as the database administrator and data migration technician (d) four part time programmers per semester. The whole team was part of ITB's *civitas academica*, who understood the context, environment and had experience in the business process.

During the first year of development, the team was placed under the vice dean's special task force, while in the second and third year, the project was sub-contracted to the business unit of the university. Most of the original developer team went on to take part in sub-contracted team joint by some fresh incoming programmers for each work package. The development team was highly motivated for developing its own system to support the ITB change management.

3.1.3 End user

We determined four categories of end-users: (a) students, as the largest quantity of user community (b) operators at the central level, as data entry for centralised operation (c) administration data operators to enter data in distributed location, and (d) executives, who have to make decisions based on the collected data and derived information.

We found it was very hard to change the work culture of many user groups, particularly the middle management level. Most of middle managers lacked of experiences for working in computer-networked environment and relied on their subordinate for information and report. However, ITB top executives demonstrated a good commitment in the implementation phase of the new system to achieve the best result for the academic business processes. They used the new system intensively providing a good example for their subordinates at ITB.

We also found that the success of the new system lies in the hand of administration personnel, which had limited experience and knowledge for working with computer. Fortunately these users were willing to learn and were motivated by the benefits of the new system. Their involvement in the development process -especially in the data cleansing process and software testing- has been very constructive. End users also play a significant role as software testers. They provide suggestions for next release and provide feature requests for future versions. Students proved to be very capable users based on their familiarity with the processes at the university.

3.2 Business Process Definition

Business process definition is the heart of our software. We identified and documented ITB's main business processes and group them into ten packages, which later approved by Vice Rector for Academic Affair.

The system must support the academic life cycle described in these packages: curriculum set up, student plans, class registration, course scheduling, students registration, course add/drop, grade recording and verification, student interim status, early warning system, student termination (graduation, drop out, withdrawal), transcript, certificate, and alumni. All academic activities are run under the responsibility of the Directorate of Education, which acts as the centralised registrar office, following a single academic calendar system.

The formerly centralised registration process moved to a distributive process gradually. Today, students no longer have to come to the booth of Directorate of Education and stay in long line, they are now able to carry out the administration process and take the registration book in their own Study Program.

3.3 Software Project Management and Development Process

Our work confronted complex issues such as processing of a high-quantity of data, variation of users, and redefinition of business process. Thus we proposed a three year plan, with clear targets and deliverables. We also control and monitor the project on a daily basis under well defined guidelines. Our works started with top-level business process definition and documentation before designing the work packages.

In brief, our plan consists of:

- Global database design, incremental table creation inline with work packages.

- Incremental development per work package, which consists of detail design (user interface, algorithm and database design), implementation (coding and unit testing), migrate data (as necessary), testing with user, feedback from user, lesson learned from current package, improvement of user interface, functionalities and development methodology of the next package.

In a continuously changing situation, it is impossible to apply a strictly predictive methodology. Thus we used incremental software development. During three year timeframe, we maintained three versions of our product. The first version of public and academic staff web site is built in a "quick and as clean as possible" way, although it was not perfect. The objective was to have a subset of the system working, which can be reused later. The second version was redeveloped by introducing a common library and framework; program patterns and templates. Some functions of the first version were abandoned though. Performance improvement was made for the client server application. The third version is improvement of the second version, with new functions and executive information system developed based on metadata, and implementation of web site which has to handle 16,000 student accounts.

During this process we found that most of the operators had no previous experience of distributing online transaction although they have already had experience in using office application. The first step is to educate operators in electronic data processing using remote transaction. To facilitate communication, a mailing list was set up in accordance to the categories: *software operation*, *problem and bug reporting*, *academic problems* and *academic official staff*.

Operators were trained to use the SIX website at every module release. They were assigned several work sessions which are organized in a same location, thus encouraging the collaboration and sharing of experiences. At the same time, administration staff provides data in legacy formats and the data entry was carried out. Bugs were registered, immediately corrected followed by regression test. Feedback was noted for further improvement.

3.4 Legacy Data Migration

The size of to-be immigrated data was very enormous, as we portrayed in Table 2.

Table 2. Quantitative Fact Sheet of ITB Academic Information System

Entity	Volume	Description
Executives	30 users.	Rector, Vice Rectors, Senates, Dean, Department Head.
Faculty/School	9 Faculty/School.	One Faculty/Schools comprises many Department/Study Program.
Departemen/Study Program	33	Department is suppressed from the system.
Faculty staff (lectures)	2,000	Academic system manage lecturer data.
Student Body	16,000 students. Growth rate : 2000 students/year.	Undergraduate, graduate and postgraduate students.
Curriculum structure	50 Study Plan.	Semantic network, must be mapped to courses and grades.
Courses	6,000 courses.	General courses could be taken by any Department. Gray area courses (could be taken by undergraduate and graduate students).
Common First Year Program	First year programs : 2,000 students per year.	ITB must deliver interim report of Common First Year program.

Entity	Volume	Description
Early warning system students	1,000 students/year.	One year before drop out termination, students will get early warning, and will be monitored intensively.
Academic Transcript	1 academic transcript per students.	At any time, before graduation, legal transcript request can be submitted for any purpose. Transcript data must be kept eternally since alumni can always ask transcript.
Certificate	1 certificate per alumni.	Certificate must delivered 1 weeks after graduation
Courses Transactions & final grade.	Undergraduate students : 6-8 courses/semester Graduate students : 2-5 courses/semester. Post graduate students : unlimited (before SIX)	Maximum length of study: - undergraduate : 4-7 years (since 2006 becomes 6 years). - graduate: 2 years. - postgraduate : 5 years
Alumni Information	Growth rate : 2,000 alumni per year.	Obtained from 3 periods of graduation per years and one period of termination.

Even though ITB has been established in 1920, and computer systems were introduced in 1977, most of the academic data were not stored in digital form – most of them were stored in student register books and managed by the Directorate of Education.

Data migration might appear as a simple process in moving data from one database to another. But the objective was to have "clean", accurate and non-redundant data. In our case, source data was not integrated and consolidated. Moreover we had to redefine the business process. Thus in our work, we did not commence any cleansing activity *outside* the new system. Clear migration rules and an incremental process before data can be declared as clear.

We decided to bring the complete legacy data into the new system, and provide a special application for data cleaning with help from the users. After the data-chunks status have been confirmed by users, it was used for operational services and then approved by all level of management. Due to the huge volume, data were gradually migrated based on their priority. We also employed a full time data base administrator to control and manage this migration process.
In 2003, ITB's management decided that all data must be confirmed and approved by all level of management, and managed by SIX. In the first 2 years of operation, SIX had to manage new academic data and also "unclean" legacy data.

The cleansing process started with data of study termination (either graduated or dropped out), because academic management promptly needs this data for important decision. Once this was completed we then shifted to clean another chunk of legacy data.

During migration process, we encountered a serious problem when ITB adopted a new curriculum. Both legacy and new data have to be examined and mapped to the new curriculum. At the same time we took this as an opportunity to clean all student academic data (especially grade data). Naturally the project work-load significantly increased. Thus we had to employ additional data entry operators to help things out.

At the conclusion of our work we found, that for a huge volume data migration -where there is almost impossible to quickly migrate and clean the data -the migration process should be engaged gradually based on the priority of business process (in our case is termination of study).

We also provide a monitoring tool attached to each module in SIX. This tool has been very helpful to monitor the quality of data and gives report if any data anomaly occurs.

4 Lesson learned and Future Direction

Developing a new system parallel with data migration from legacy system is an effort consuming task. However, this paper proved that at the end we were successful with our migration project. The role of top executives was very important in the process, as they provided motivation for lower level staff. The user willingness to use the system is another determining factor. In the new system, all implemented functions proved useful and operational.

Business process definition and technical requirement engineering also played an important role for translating business definitions to software solutions.

Before implementing incremental solutions, we believe that solid software architecture definition and complete database design must be done, particularly considering the legacy data migration.

We have to maintain growing versions iteratively: feedback gathered from one package will accelerate the next production cycle significantly. As a whole result: no part of the software is of no value. This usability aspect that we obtained from an end-user driven approach also satisfies us as developers. Whilst often in the past, software was built but was never actually used, our system has been fully used by the user community on various levels of the ITB hierarchy.

Additionally it is worth noting, that the whole system was developed with Open Source Software (OSS). This has proven that OSS can be used for supporting the operation of a university information system. Before the project started we were very concerned that using OSS will need a lot of efforts and high level of budget for development, in fact: we had to spend budget for user training and change management. However we believe that this was the best decision, since OSS-based system motivates more sustainable system in the future.

Acknowledgment

This work is dedicated to SIX development team for they hard work for the last three years. We want to thank all administration staff in ITB, for their cooperation as a solid team and full support to the new academic information system. Our appreciation to ex-Rector of ITB, Kusmayanto Kadiman PhD, and Vice Rector for academic Affairs Prof. Dr. Adang Surahman, for fully supporting the development of the new system toward academic excellence of ITB.

5 References

[1] Aversano, L., Canfora, G., Cimitile, A., De Lucia, A., Migrating legacy systems to the Web: an Experience Report, Fifth European Conference on Software Maintenance and Reengineering, IEEE, 2001.

[2] Bergey, J., Smith, D., Weiderman, N., DoD legacy system Migration Guidelines, Carnegie Mellon University, September 1999.

[3] Bianchi, A., Caivano, D., Marengo, V., Visaggio, G., Iterative Reengineering of legacy systems, IEEE Transaction on Software Engineering, vol. 29, no. 3, March, 2003.

[4] Biffl S., Aurum A., Boehm B., Erdogmus H., Grünbacher P. (eds.), Value-Based Software Engineering, Springer Verlag, 2005, ISBN:3-540-25993-7.

[5] Bisbal, J., Lawless, D., Wu, B., Grimson, J., Wade, V., Richardson, R., O'Sullivan, D., Overview of legacy information system migration, The 1997 Asia-Pacific Software Engineering Conference and International Computer Science Conference, APSEC'97 and ICSC'97, Hong Kong,1997
[6] Bisbal, J., Lawless, D., Wu, B., Grimson, J., legacy information system: Issues and Direction, IEEE Software September/ October 1999.
[7] Boehm, B. W., Software Engineering Economics, Prentice Hall, 1981
[8] Boehm, B. W., Value-Based Software Engineering, Software Engineering, Notes 28(2), pp 1–12,May 2003
[9] Brodie M., Stonebraker, M., Migrating legacy systems: Gateways, Interfaces and the Incremental Approach, Morgan Kaufmann Publishers Inc. 1995.
[10] Freeman, R., E., Strategic Management: A Stakeholder Approach, Pitman Publishing Inc., 1984.
[11] Ganti, N., Brayman W., Transition of legacy systems to a Distributed Architecture, John Wiley & Sons Inc. 1995.
[12] Preston, L., Donaldson, T., Brooks, L., Principle of Stakeholder Management, The Clarkson Centre for Business Ethics, 1999.
[13] Wu, B., Lawless, D., Bisbal, J., Richardson, R., Grimson, J., Wade, V., O'Sullivan, D., The Butterfly Methodology : A Gateway-free Approach for Migrating Legacy Information Systems, Third IEEE International Conference on Engineering of Complex Computer Systems (ICECCS '97), 1997.