KEY FACTORS IN LEGACY SYSTEMS MIGRATION
A Real Life Case

Teta Stamati, Konstantina Stamati, Drakoulis Martakos
Department of Informatics and Telecommunications, University of Athens,Panepistimiopolis, Athens, Greece

Keywords: Legacy systems, migration methodology, drivers, hinders, critical success factors.

Abstract: Although legacy systems migration as a subject area is often overlooked in favour of areas such as new technology developments and strategic planning of information technology, most migration projects are considered ill-fated initiatives and a rate of over 80% of these projects run over budget, frequently with system functionality falling short of contract. Many practitioners consider that the proposed theoretical migration approaches are myopic and do not take into account a number of key factors that make a migration project a really complex initiative. Our position is that throughout the life cycle of a migration process, there are some critical factors that initially play the role of the "drivers" and afterwards they became the factors that hinder ("hinders") the migration process. We consider these key factors as Critical Success Factors (CSF) that must be carefully considered. Furthermore, these key factors could be either overt or covert factors. In each case, the migration engineers should consider and analyse them very carefully prior to the initiation of the migration process and a well-defined migration methodological plan should be developed. The work presented is based on a real life initiative putting emphasis on the key success factors revealing at the same time the complexity of a migration process. Emphasis is put on the required management view and planning effort, rather than on the mere technological issues.

1 INTRODUCTION

The term "legacy system" is one of the most widely used and somewhat misconstrued terms in the industry today. It describes an old system that remains in operation within an organisation and often represents a massive, long-term business investment that could be composed of extremely efficient robust and fine tuned applications built over many years by a combination of IT and business experts. Moreover, the term "legacy" is used in a pejorative sense to describe applications based on old and obsolete technology. In general, the legacy applications represent the storehouse of business intelligence to support and manage core business functions and provide critical information for day-to-day operations. In fact, legacy applications, especially those based on mainframe platforms, are the mainstays of today's businesses.

Several alternative definitions of what exactly a legacy system is can be retrieved. Ulrich (1994) defined them as "stand-alone applications built during a prior era's technology, but they are perhaps more widely understood as software systems whose plans and documentation are either poor or non-existent (Connall and Burns, 1993)". Bennett (1995) referred to the legacy systems as "large software systems that we do not know how to cope with but that are vital to the organisation", while Brodie & Stonebraker (1995) as "any information system that significantly resists modification and evolution to meet new and constantly changing business requirements". In all the aforementioned definitions, often the notion of "something valuable" is present, as well as the notion of "old and obsolete" (Lauder and Lind, 1999). It is unambiguously recognised that legacy systems are crucial for the operation of organisations and thus they are essential for our economical and well-fare activities.

The commonsense solution for an organisation to the legacy problem is migration of its mission critical legacy systems. The definition of a successful information system migration according to Brodie & Stonebraker (1995) is as follows: "it begins with a mission-critical legacy system of a significant size in full operation and it ends with a fully operational, mission critical target application (or applications components) that replace the

Stamati T., Stamati K. and Martakos D. (2006).
KEY FACTORS IN LEGACY SYSTEMS MIGRATION - A Real Life Case.
In *Proceedings of the Eighth International Conference on Enterprise Information Systems - DISI*, pages 340-344
DOI: 10.5220/0002469403400344

essential aspects of the original legacy system". This involves replacing the problematic hardware and software, including the interfaces, applications and databases that compose an information system infrastructure. Brodie & Stonebraker (1995) claim that legacy information system migration involves starting with a legacy information system and ending with a comparable target information system. This target system is significantly different from the original, but it contains substantial functionality and data from the legacy system.

Within this context, the main objective of the present work is to refer to a specific case of a mission critical migration initiative revealing some significant factors that should be seriously considered by both migration engineers and business experts during the life cycle of a legacy systems migration initiative. The case described is based on a spin-off migration project for the Supply Chain Management (SCM) Department of a Greek Public Organisation (hereinafter referred to as "the Organisation").

For reasons of confidentiality, there is no reference to the name and nature of the Public Organisation as well as some details in this experience are made somewhat more generic, but the conclusions remain sufficiently based on a real experience to be relevant in some contexts.

2 DRIVERS AND HINDERS AS CRITICAL SUCCESS FACTORS

The main mission of the Organisation's SCM Department is the general support of all the in-house supply chain management processes, their provided services as well as the satisfaction of requirements of distinct Units within the Organisation.

2.1 The Context of the Case

In the beginning of 1999 the Organisation published a Request for Proposal (RFP) initiating the evaluation procedure of outsourcing the migration process of its mission critical mainframe system to a new architecture. The RFP requested the most effective and efficient methodology in order to transfer the legacy functionality (legacy system) to a new environment (target system). The case was mainly driven by the need to revitalise the Organisation's IS infrastructure that was crucial for the SCM Department and focused on the need to reengineer its large scale legacy environment. More specifically, the objective was twofold: the "2000 problem" resolution as well as the development of a new information system that would include all the existing operational functionality as well as the new operational requirements that had been raised by the obligation of conformity with the introduced logistics regulation for the Greek Public Sector.

The duration of the tenders' evaluation process was four months and an integration company (hereinafter referred to as "the Integrator") was selected among several candidates as the most appropriate company to implement the project.

The main argument that the Integrator had used in order to gain the tender was that the company had already developed a well-defined methodological approach called "Bridge Migration" method in order to successfully implement prior migration initiatives mainly in the Banking Sector.

The "Bridge Migration" approach was based on the use of gateways in order to successfully handle the transformation from the legacy system to the new environment. A robust combination of the state of the art methodologies for successful migration had been included within the Integrator's technical proposal and based on its proved know-how, the company promised to the Organisation the immediate customisation of the "Bridge Migration" approach to the specific needs of the Organisation. The main principles of the approach were as follows:

- transfer all the legacy data activities to the target within a narrow time frame acceptable by the legacy system.
- restart/recover after a system failure all of the data in all technology platforms involved.
- reconcile and account of data and immediate availability of the data to the users of the corresponding applications.
- incorporate into the system the auditing functionality on data transferred and system status reporting.
- handle processing of the errors, exceptions and rejections, stemming from referential integrity constraints, data errors and/or technical reasons. Individual rejected transactions would be selectively resubmitted for processing or dropping from the system.
- consider the inter-dependencies between the applications and provide a mechanism to service these dependencies during operation.
- provide priority levels on data transfers, so that emergency updates and back-log bypass would be allowed.

- facilitate the roll-out of the target system and the shut-down of legacy system.
- include tools for the operation and monitoring of the system. Parameters governing functionality, such as start/stop times or data commit frequency, could be integrated within the system. The legacy system could dynamically set these parameters on an individual file interface level. Also, the one-to-many routing of data from target to legacy would be table driven and dynamically determinable.
- include an interface for correcting data inconsistencies resulting from lack of synchronization between target and legacy. It could as well report on such off-line events.
- apply a large volume data changes, such as batch updates not depending on system's operation and availability.

The Integrator's underlying philosophy of implementing the project was that the engineering team involved in the project had the know-how to successfully implement migration projects and it could make the specific approach fit to the Organisation's requirements.

2.2 Drivers for Legacy Migration

The Organisation's SCM Department had been involved in the in-house development of the legacy system for more that thirty years. The system formed the central hub and the backbone of the information flow within the Department and was the main vehicle for consolidating information about the supply chain management processes.

The old system was built in COBOL with an obsolete Command Line Interface and the data were stored in Index (flat) COBOL Files.

The Organisation's legacy system was posing numerous and important problems to the host Department. Some of the worst and lamentably typical key characteristics that played the role of the "drivers" in the *system level* for having the system migrate and evolve were the following:

- it was too large, with millions of lines of code. Furthermore, it was too old (more than thirty years old).
- it was written in COBOL which was considered a legacy language. The later resulted in a situation where there was a significant loss of institutional knowledge as the original developers were retired. Furthermore, the Department was facing a skills shortage as the academic institutes had already stopped teaching the COBOL.
- it was built around a legacy database service namely, flat-files.
- it was running on obsolete hardware, which was slow and expensive to maintain.
- the growing business volume and the data processing model used made the system performance an increasingly important issue.
- maintaining the software of the legacy system was an expensive task and the process of finding and correcting system faults was also costly and time consuming because of the significant lack of documentation and a general lack of understanding of the internal functionality of the system.
- lack of clean interfaces brought obstacles in the integration process with other systems.
- it could not evolve to accommodate new functionality that was required by the Organisation.

To complicate matters, the system was considered as mission-critical and had to be operational at all times. It was difficult to modify the system while it continued to perform its mission-critical functions.

Additionally to the aforementioned "drivers" as far as the system was concerned there was as well a significant number of *business "drivers"* for having the system migrate, as follows:

- the Department's attempt to redefine its strategy from a traditional data processing model to a multi-channel, service oriented model. This enforced the requirement to have up-to-date data, coming from multiple sources, on-line at all times. It should be noted that the data was often replicated at local offices, for historical reasons and the historical data was often only available off-line, e.g. archived on tape.
- the coming of the year 2000 would generate to the system the well-known problem of the dates.
- legal requirements and regulations in Greece for the Public Sector were under investigation and potential changes hardly ever consider the IT system characteristics.

2.3 Hinders of Legacy Migration

The initial timeframe of the project prescribed the completion of the whole initiative within less than eight months. This short period of time for such a complicated project had been considered as an imposed constraint that the project team had to adhere to. Due to that short project life cycle the Department considered that there was no time and need to produce from scratch the functional requirements for the part of the applications that would replace the legacy functions. The main underlying argument of the project steering committee was that ultimately the target system should have exactly the same functionality as the old one as far as the existing[1] functionality was concerned.

The combination of the millions of lines of COBOL code with the lack of documentation and comments within the code proved to be a serious obstacle for the in-depth understanding and analysis of the code in order for the involved team to obtain the detailed knowledge of the existing functionality. This was the main reason that the ultimate required man-months for completion of the migration process were much more than the initial estimation of the man-effort within the project plan. Considering the aforementioned "hinders" factors, the project steering committee decided to proceed to the following scenario: separation of the whole code in functional groups, removal of the redundancies and keeping the core applications blocks in COBOL. Afterwards, the engineering team built new functionality around these core COBOL applications and linked them with a new relational database and windows-based interface and reporting tools.

The project delivered to the Organisation after eight moths and the project's committee considered the new system as a fully operational and successful system. Thus, although the new target system consisted of some "old" parts of the legacy system (COBOL applications) preventing potential further enhancement of the target system, the project's steering committee decided that the new system covered their current needs.

3 CRITICAL SUCCESS FACTORS OF LEGACY MIGRATION

Due to the complexity that a migration initiative reveals, a well-defined methodological action plan is required (Stamati et al., 2004). This plan should consider in detail all the CSFs which could play initially the role of the main motivators that drive the migration need and afterwards could be constraints that hinder the same process. Thus, it is necessary to produce a well designed plan of the whole process as far as the methodological perspective is concerned. The technology drive should be present, but should not be overestimated.

In our case, a more careful evaluation of the "as-is" situation in the initial mainframe environment was required. Relevant action items were among others the evaluation of the pre-existing COBOL code and the underlying data structures. Determination of the "to-be" business and application architecture along with the corresponding "to-be" technical and deployment architecture were essential for the completion of the project. Furthermore, critical evaluation of the migration scenario between the "as-is" and the "to-be" situations should have been further analyzed and designed in depth.

Each migration process must be gradual (Holland and Light, 1999). We should not proceed to a decision of a big band migration or to a short period initiative. This could add constraints to the decision that narrow the ultimate result of the system.

Finally, the success of a migration effort has to be measurable (Sommerville, 2001). In our case, it is noteworthy that the whole project plan was continually adapted considering the new obstacles and constraints. Although the project's steering committee was fully satisfied with the target delivered system, the lack of a well defined list of CSFs that would have been used to evaluate the process should have been considered as essential for the final validation of the system.

Therefore, a number CSF's must be defined. Those should not only be used to evaluate the final outcome, but should be used along the road to measure progress and to define decision points in the plan, where go/no-go decisions must be made.

In this particular case, the following indicative CSF's could identify that:

- a significant proof of concept must be delivered within a narrow amount of time and effort
- along the whole roadmap, quick wins must be identified to validate the migrating system, and those must offer a measurable business benefit
- support from the stakeholders must continuously be ensured

[1] The Department provided to the Integrator the requirements' handbook as far as the new additional functionality was concerned.

- from management
- from users, both business and regular

Generally, the CSFs should mirror the successful implementation of a migration project. The CSFs during a migration initiative must ensure that the requirements and specifications should be completed, the assertions must be testable and detailed requirements must be done jointly. Executive support, clear vision, well-defined business objectives, good project management, definition of a formal process, firm basic requirements, skilled staff and user involvement are considered as imperative factors for the success of such projects. Acceptance criteria should be initially defined considering the required functionality, the test cases and the performance targets. User management commitment is essential as well. Managers should be results-oriented, act as internal champions for the migration project and be committed to a user/developer partnership model. Finally, quality assurance and performance metrics should be considered as key factors.

4 CONCLUSIONS

Any effort for migration or evolution of large-scale industrial software systems must be driven by a corporate strategy redefinition. Buy-in from management and end users requires a well-defined strategic management view and planning, based on a clear statement of motivation, objectives and of the constraints imposed by the current environment and continuous operational needs. Thus, any migration process must be implemented as a planned change process that first and foremost requires an understanding of the range of issues and organisational entities involved. The definition of measurable success factors is essential. The organization-wide methodology must be defined and supported by all stakeholders.

ACKNOWLEDGEMENTS

Much of the work presented in the sections above was undertaken under the research project called PYTHAGORAS: Help of Research Teams in the Universities, that is part-financed by the Operational Program of Education and Initial Professional Training-EPEAEK and the European Fund-EKT.

REFERENCES

Bennett, K.H., 1995. Legacy Systems: Coping With Success. In *IEEE Software*, 12(1), 19-23.

Brodie, M.L. & Stonebraker, M., 1995. *Migrating Legacy Systems,* Morgan Kaufmann Publishers. London, 2nd edition.

Connall, D. & Burns, D., 1993. Reverse Engineering: Getting a Grip on Legacy Systems. In *Data Management Review*.

Holland, C., P. and Light, B., 1999. Focus issue on Legacy Information Systems and Business Process Change: Introduction. In *Communications of AIS*, 2(9), 98-120.

Lauder, A. and Lind, M., 1999c. *Legacy Systems: Assets or Liabilities?*, In LAP'99, Copenhagen.

Sommerville, I., 2001. *Software Engineering*. 6th Edition, Pearson Education.

Stamati, T., Kanellis, P., Stamati, K., Martakos, D., 2004. *Legacy Migration as Planned Organizational Change*. In Sixth International Conference on Enterprise Information Systems (ICEIS) 3, 501-508.

Ulrich, W., 1994. *From Legacy Systems to Strategic Architectures.* In Software Engineering Strategies, 2(1), 18-30.