Legacy System: Migration Strategy

Ashok RB Samuel

About Technowave, Inc.

Technowave is a strategic and technical consulting group focused on bringing processes and technology into line with organizational goals. Technowave provides best of breed solutions to real world problems for the Small to Medium Business (SMB) market. In addition to strategic consulting, Technowave provides system architecture, project management, development and implementation services. Technowave also provides technical and knowledgeable professional with Innovative Technologies to organizations on a flexible, contract and permanent basis focusing according to the client's dynamic needs.

For additional copies of this whitepaper, or to explore partnership opportunities with Technowave, please contact us as follows.

Technowave Inc.
900 Pump Road, Suite 69
Richmond , VA 23238
Tel: (804) 740-0957
Fax: (804) 740-0958

Information and communication technology is in a rapid and dramatic flux of change. However, legacy systems are not influenced by these changes due to their size, complexity, and importance. It is a great challenge for IT managers to decide whether keep or retire them because the bulk of a company's business processes is locked up in these legacy systems. This section of the paper discusses some distinct characteristics of legacy systems, and some successful migration and integration strategies.

Supported Software and hardware

Based on the free on-line dictionary of computing, Legacy system is a computer system or application program which continues to be used because of the prohibitive cost of replacing or redesigning it and despite its poor competitiveness and compatibility with modern equivalents.

A programming language called COBOL, one of the few things that have been around in the IT industry for more than 40 years, was used to develop most of the legacy systems. Cobol is the one of the third generation high-level language that facilitates developers to write code in symbolic instructions. Cobol was designed to write English-like syntax, which is not only readable by programmers, but also by the managers and business people. In a mainframe environment, DB2 is the commonly used database with Cobol. Cobol provides the SQL (Structured Query language) standard interface for database access. Cobol also supports a hierarchical database such as the IMS database system developed by IBM. Fortran and Pascal were also used to develop some of the legacy systems. Most of the legacy system was developed with IMS or IDMS databases. In the Hierarchical model (IMS), a group of fields is called a segment instead of a record. Each segment can have only one “parent” but more than one “child”. To access a data from any data segment, the user needs to identify the root first, and then follow the branches until the child data segment is found. IDMS is the popular database from the Network model or CODASYL DBTG model. The network model contains a tree structure like the Hierarchical structure, but can include several trees with shared branches within the same structure.

Most of the legacy systems were developed on different hardware architectures, including the PCs, mid-range computers, and mainframes. Programs can be compiled and run on variety of operating system such as DOS, AS/400, VSE, VMS, VM, and MVS.

Advantages and Disadvantages

Based on the book "Legacy Systems: Transformation Strategies" by William M. Ulrich, the following are the major disadvantages and advantages of the legacy systems.

Disadvantages:

- Difficult and time intense process to understand the system functionalities.
- Hard to find and separate business logic from presentation and data logic.
- The same business logic can be used in multiple systems with different versions of implementation.
- Hard to find functional or technical documentation.
- Difficult to integrate with current system because of hardware and software incompatibilities.
- Hard to enhance or modify the systems, since the documents and regression tests are not available.
- Hard to manipulate and retrieve data because of the redundancy.
- IMS and IDMS database system do not support relational data base architecture, which affects data integrity and consistency.

Advantages:

- Able to support existing business process. The Cobol Report stated that CICS transaction volume (such as COBOL-based ATM transactions) grew from 20 billion per day in 1998 to 30 billion per day in 2002.

- Proven and reliable. Most of the business functionality have been built and proven. The Gartner Group reported that 75% of all business data is processed in COBOL. There is no need to invent any concept from scratch.
- Cheaper to maintain because legacy systems are scalable and robust.

Migration Strategy

We can either eliminate or integrate the legacy systems. To eliminate or integrate a legacy system, we should follow these steps:

1. The impact analyses for the targeted process are reviewed to identify the scope of the elimination or integration effort.
2. If the elimination is an iterative effort over multiple releases, the components targeted for elimination in the current release are identified. If the elimination is not iterative, proceed to step 4.
3. Identify the components for interim integration in the current release.
4. Identify broken links, reports, and processes to rebuild as a result of the elimination and for interim integration.
5. Identify the data conversion scope.
6. Identify user impacts and any user system conversion required; e.g., modified data submissions from external users.
7. Identify potential workarounds that may be implemented to mitigate impacts of the elimination effort.
8. Review and update the cost/benefit analyses for the elimination/mitigation alternatives; e.g., compare alternative data conversion strategies.
9. Develop the project plan for the elimination effort. This step includes the decisions on which strategies to employ for the elimination effort.
10. Develop the detailed data conversion strategy and data conversion plan and design.

11. Develop the detailed plans and designs to rebuild broken links and reports.
12. Develop detailed test plans (building test cases and scenarios are developed in the elimination plan execution step).
13. Develop detailed production closeout plan. This plan includes all changes to run programs, copy libraries, job streams and security profiles that are impacted by the elimination effort, as well as plans for renegotiation of maintenance contracts, retirement and archival of configurable items for emergency access to historical data and processes, and data retention and archival as specified in requirements, including legal requirements and corporate policies. These closeout plans should ensure that retired legacy system applications, software and historical data are secure from unauthorized access after system elimination, and should specify procedures for the authorized retrieval and restoration of legacy configurable items and historical data.
14. Develop the staff transition plan for project and support staff.
15. Finalize the project plan for the elimination or integration, incorporating the plans developed in steps 10 – 14.
16. Execute the elimination or integration plan.
17. Eliminate or integrate the legacy process or system, and perform all production closeout activities.

<u>Conclusion</u>

Legacy systems are still alive because of their distinct characteristics and good pedigree. In the last 40 years, we have learned that it is neither practical nor affordable to migrate $5 trillion worth of legacy code into other technologies which were short lived. However, it is possible to either eliminate or integrate the legacy systems by following effective migration strategy and appropriate migration tools.