Cost of Supporting Legacy Systems

Introduction:

A legacy system is a system that was written 10+ years ago using what is now perceived as 'Old' technology and methodologies. For instance when legacy systems are mentioned, COBOL immediately comes to mind, together with monolithic, spaghetti code.

These systems are still used as they still meet the business requirements and needs even though there are more efficient technologies and methods available.

Within the code and data stores, legacy systems contain a wealth of information that could be used to support new strategic business initiatives.

Since software systems don't wear out, legacy systems are viewed as old and creaky, but are tried and true and 'meet' current requirements. The realization of the need to modernize may hit long after the time when the transition may have been made easily and relatively risk-free.

The reality is that legacy systems are risky for the businesses that run them, particularly strategic business risks associated with their costs and inflexibility. These systems may have been in production for 20-30 years they have been stable, but inflexible and made brittle over the years by ad-hoc maintenance and enhancements. This inflexibility and instability make legacy systems difficult and expensive to maintain. Up to 60-80% of IT budgets are spent on maintaining legacy applications.

Fragility:

Systems that have been maintained, enhanced and patched over a number of years become unstable and fragile. Developers become reluctant to make changes as they are uncertain what side effects may be caused by these changes. Documentation has become so out of date to be effectively worthless and developers have to dig into the code to try and understand where the changes need to be made without affecting anything else. Often there are sections of code that due to the enhancements and patches, are never executed, further adding complication to any enhancement initiative.

Reverse-engineering your code will identify all orphaned code, allowing you to update or create a new document set. Developers will be confident in making any changes required, reducing the cost and risk of any maintenance or enhancement initiative.

Flexibility.

Being able to respond to business or legislative changes quickly is a necessity in the current economic climate to remain competitive. Unfortunately, since legacy systems were built based on 10-20 year old technology, the ability to react to these changes quickly is limited.

What is this costing you in:

- Reduction of overall productivity.
    - Do you need to create manual workarounds to compensate for system limitations?
- Limiting Competitiveness.
    - Does the system fail to provide adequate or timely business data?
- Lack of Qualified resources
    - Many of the SME (Subject Matter Expert), both technical and business, are approaching retirement, and it is difficult to find younger qualified personnel. COBOL is no longer taught and is not a 'sexy' language for the younger generation to learn. They see it as a career limiting option.

Building a road map will help you move your legacy systems into the future, enabling you to better react to the changing business environment, allowing you to remain competitive.

Growth:

Does your legacy system:

- support increased capacity due to business growth?
- take advantage of the WEB or transition to a web-based application?
- have the capability to export/import data to/from other systems?
- Provide easy access to timely business data?

Up to 60-80% of IT budgets are spent on maintaining existing systems. This leaves only 25% of the budget for new business initiatives. Reducing maintenance costs frees up more resources, both technical and financial, to address new strategic projects.

Business Continuity:

- Would your company be able to continue to operate if the legacy system fails?
- Is replacement less expensive than paying for support?
- Is the legacy hardware supported? – Replacement parts available?
- Are qualified resources easily available to maintain and support the legacy systems?
    - For instance COBOL is no longer taught and is not a 'sexy' language for the younger generation to learn. They see it as a career limiting option.
- Does the legacy software only run on antiquated hardware?
- Is there sufficient documentation available to support the system once the SME has left?
    - With ad-hoc maintenance and enhancements over the years, the documentation probably is not up to date, even if available.

Conclusion:

Legacy systems contain a wealth of information to run your business. Strategically planning to move legacy systems into the future will mitigate the risk of continuing to support your business and allow you to react quickly to any changes in the business environment.