PLAN PROPERTY FOR THE CRITICAL UPGRADE
By Al Moore
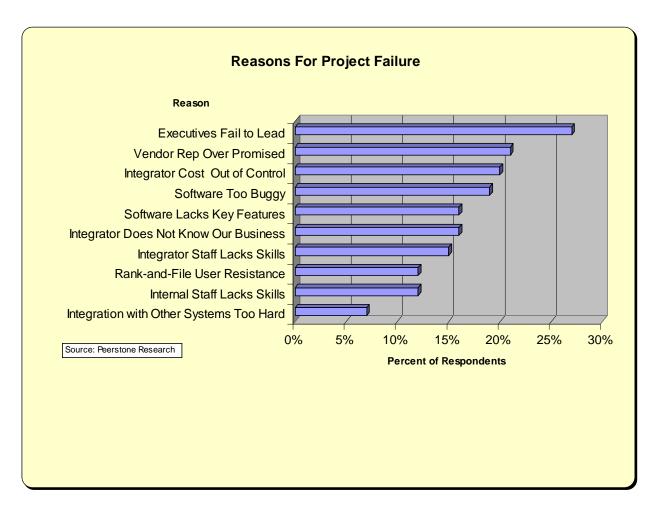**Marathon Consulting, LLC.**

Part I

About the Author: *Al Moore is a founder of Marathon Consulting LLC., a Virginia Beach-based Information Technology services firm specializing in Custom Software Development and Technology Advisory Services. Mr. Moore has over 25 years experience in the IT profession, spending the majority of that time helping clients leverage technology to achieve tangible business results.*
Contact Info: 757.641.2291 or, amoore@marathonus.com

Now is the time of year when business leaders think about enterprise-wide system replacements. Conventional wisdom says, "If we start now, we can have it done by the end of the year". This is valid in many cases. However, if not undertaken with a great deal of care and commitment, these projects can have extremely unsatisfactory and costly results.

Replacing a mission-critical information system is one of the most risky projects an organization might attempt. Failing to define and understand business needs, making a faulty software choice, or failing to execute a smooth implementation can drive a healthy organization into despair. Even minor glitches can cause embarrassment and a loss of confidence in your organization. Serious problems often result in lost revenue, customer defections, employee turnover, and costly operational interruptions. A successful system replacement should be invisible to your customers and trading partners. With implementation costs averaging from three to seven times the purchase price of the software, "Do-Overs" are simply not an option.

The risk of failure is large and well documented. Various studies conducted by industry watch groups are not encouraging. One of the most cited surveys, conducted by the Standish Group and updated frequently, reveals that approximately 31% of all software projects are cancelled before completion. Further, the most common size projects, those costing between $700,000 and $2,000,000 experience success rates between 18% and 55%, at best.

Another study from *Peerstone Research* focuses on implementation projects like those addressed in this article and the most often cited reasons for failure. This research tells us that the number one reason is that senior executives fail to lead. Six of the nine remaining reasons focus on shortcomings on the part of the software or services vendor.

Reasons For Project Failure

Reason

Executives Fail to Lead	
Vendor Rep Over Promised	
Integrator Cost Out of Control	
Software Too Buggy	
Software Lacks Key Features	
Integrator Does Not Know Our Business	
Integrator Staff Lacks Skills	
Rank-and-File User Resistance	
Internal Staff Lacks Skills	
Integration with Other Systems Too Hard	

0%   5%   10%   15%   20%   25%   30%

Source: Peerstone Research

Percent of Respondents

Every practical step must be taken to mitigate those risks identified above. The following series of articles outline a set of rules that an organization can follow in order to avoid these pitfalls. This article addresses the first phase of a three-phase process: Defining Your Business Requirements. It presents four practical rules that will help you reduce these risks to a manageable level.

Subsequent articles will address the second and third phases: Evaluation of Software Solutions and Implementation Management.

Rule Number One - Insist Upon Consistent Management Commitment and User Involvement

The motivation for the project will come from either the business/operational side of the organization, or the technology side. Regardless of the source, management and user involvement is mandatory throughout the project. Helpful techniques include:

1. Defining how much time each particpant can dedicate to the project

2. Explaining to the management team that the overall success of the project is their shared responsibility – they will own the system and must make it work

3. Having the senior executive consistently express his or her commitment to the project

4. Conducting scheduled status meetings – identify and resolve resource issues

Rule Number Two - Know Your Business Requirements

The Standish Group noted that over 35% of project failures were caused by incomplete requirements definition, changing requirements, and unclear expectations. Our experience supports this finding. Keys to success include:

1. Identifying all project stakeholders and ensuring they participate in the requirements definition

2. Evaluating current procedures and defining what future operational procedures will be – the "To-Be State"

3. Using a consistent methodology that identifies specific requirements on a departmental, business function basis

4. Describing specific requirements in unambiguous terms such as, *The ability to retrieve a customer record based upon Telephone Number"*

5. Documenting all findings - basic functionality must be included in the list; assume nothing about the future software's capabilities

6. Aligning requirements with strategic plans that might only be known to senior executives – acquisitions and new product lines can pose problems if not considered early

7. Ensuring that all requirements are communicated to the stakeholders and common expectations are established

8. Developing a consensus on requirement priorities

9. Identifying and resolving organizational issues that could impact the project

There is no "silver bullet" software package that provides a 100% match to everyone's requirements. Functionality tradeoffs must be negotiated in the overall interests of the organization. No further action should be taken until consensus on the requirements and relative priorities is achieved.

Rule Number Three - Set Realistic Expectations

Establish a realistic balance among system functionality, budget constraints, and your organization's capabilities.   Some keys to success include:

1. Do not over-buy by acquiring a system with features you will not use in the next three years – extra functionality brings extra costs and extra complexity, even if it goes unused

2. Stick to the defined requirements as a foundation for your expectations

3. Understand and communicate the non-monetary investments of effort and stress that will impact your employees during the implementation

Rule Number Four - Know When to Get Outside Help

The purchase of an enterprise-wide, highly integrated software package is not done often. Most organizations will not have an employee with this experience on staff. If so, it is unlikely that this employee will have nothing else to do for the next 6 to12 months. Any of the following conditions should prompt you to consider outside help. They are:

1. Your staff lacks the skill, experience, or time to lead a high-risk project requiring strong business skills, knowledge of software applications, and excellent communication and negotiation abilities

2. You are being overrun by vendors, all of whom offer viable solutions – the "Vendor Feeding Frenzy"

3. The process is likely to become hampered by office politics

4. Highly placed individuals are already identifying and lobbying for their favorite vendors

5. You have selected a solution and need an unbiased evaluation and validation of that decision

NEXT INSTALLMENT: Part II - Identifying and Evaluating Vendor Software Solutions