

Successful ERP Implementation Requires Planning

Presented At:



Presentation Number T-127
Thursday March 2nd, 2000

Author and Presenter:

Robert W. Starinsky
Managing Principal
Tradewinds Group, Incorporated
Consultants to Management
www.tradewindsgroupinc.com

rwnstarinsky@tradewindsgroupinc.com

TradewindsGroup
Tradewinds Group, Incorporated
Consultants to Management
Post Office Box 3601
Oak Brook, Illinois 60522-3601

Copyright (c) 2000 by Robert W. Starinsky.

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the author.

The author acknowledges all registered trade names, or trademarks as they may appear in this document.

Tradewinds Group, Incorporated is pleased to make this white paper available to interested parties at no charge. This paper may be distributed freely for non-commercial, academic purposes.

Third Printing, Spring 2003.

Table of Contents

TABLE OF CONTENTS	3
INTRODUCTION	5
WHY ERP PROJECTS FAIL	6
WHY COMPANIES BUY ERP SOFTWARE	7
SELECTING THE RIGHT ERP SOFTWARE	8
BEST PRACTICES AND ERP SOFTWARE	10
EVALUATING ERP SOFTWARE	11
SELECTING AN IMPLEMENTATION CONSULTANT	14
BUSINESS PROCESS RE-ENGINEERING AND ERP IMPLEMENTATION	18
THE SOFTER SIDE OF AN ERP IMPLEMENTATION	20
PROJECT SCOPE	23
PROJECT RISKS	24
STEPPING THROUGH AN IMPLEMENTATION APPROACH	25
A TYPICAL ERP IMPLEMENTATION WORK BREAKDOWN STRUCTURE	27
A BEST PRACTICE BASED ERP SOFTWARE IMPLEMENTATION APPROACH	28
CHARTER THE PROJECT	29
SELECT THE PROJECT TEAM	29
EDUCATE THE PROJECT TEAM	29
MODEL THE BUSINESS	29
CONFIGURE THE SOFTWARE	30
PROTOTYPE BUSINESS PROCESSES	30
IDENTIFY FUNCTIONALITY GAPS	31
DEVELOP TECHNICAL SOLUTIONS	31
DOCUMENT CONFIGURATIONS, CUSTOMIZATIONS, PERSONALIZATIONS AND PROCESSES	31
ESTABLISH PRODUCTION READINESS	31
TRAIN END USERS	32
GO LIVE	32
ESTABLISH STEADY STATE	32
PERFORM PERIODIC HEALTH CHECKS	33
COMPLETE PERIODIC SOFTWARE UPDATES	33
IS AN ERP IMPLEMENTATION EVER REALLY COMPLETE?	35
SOME THOUGHTS ABOUT TECHNOLOGICAL INFRASTRUCTURE	36
A FINAL THOUGHT ON APPROACHES	39
AN ERP BIBLIOGRAPHY	40

AN ERP VOCABULARY41
ACKNOWLEDGEMENTS46
ABOUT THE AUTHOR/PRESENTER47

Introduction

Much has been written in the trade press of late about the failure of enterprise resource planning (ERP) systems to deliver on their promises of true business value. Cost overruns, significant gaps, missed deadlines, marginal end results and little, if any overall business process improvement make it clear that managing an ERP project is a complex feat.

These systems generally represent large-scale investments of both dollar and human capital. Because of their organization-wide scope and duration, risks in the typical ERP implementation are significant. Therefore, the defining element in the success or failure of an ERP system implementation is largely one of project management. The presenter draws on observation, personal experience and research to build a best practice based strategy, or approach that will lead to a successful ERP system implementation experience.

Why ERP Projects Fail

Over the course of time I have seen ERP projects fail for the following reasons:

- The wrong software was selected.
- The software does not work as advertised.
- Too many changes or interfaces to the software were attempted.
- Lack of executive sponsorship for the project.
- Lack of a project plan.
- Poor or no training was provided.
- Lack of business or technical resources committed to the project.
- Lack of effective change management strategies.
- A business merger terminates a work in process.

Throughout the remainder of this paper I will attempt to share through my experience, observation and research various strategies that offer you the best chance of achieving a successful ERP implementation experience.

As a launch point for these remaining discussions here is what I consider a short list of critical factors toward achieving a successful ERP implementation:

- Select the right ERP software
- Manage expectations
- Exploit the best business practices built into the software
- Get the softer side of implementation right
- Mitigate risks
- Take a managed approach

It is also important to point out early on in any discussion about achieving ERP implementation success some important realities about ERP implementations in general. These include:

- The perfect ERP system *does not exist*.
- No two ERP implementations *are ever the same*.
- No ERP implementation *is ever perfect*.
- No ERP implementation *is ever complete*.

Why Companies Buy ERP Software

Companies can't be good at everything. In an increasingly complex, global, highly competitive and time sensitive business climate, companies must more than at any previous time, focus on and exploit core competencies. In short, companies today must achieve outstanding performance in their core businesses.

Is your company a software factory? If not, then building sophisticated, integrated information systems is not the core, or main business of your company. It is for this reason alone that any internally built information system will generally not equal the quality, scope or technology employed in a system created by a business whose core business is the creation of sophisticated, integrated information systems.

Selecting and implementing the most appropriate software package to support a company's business processes is itself considered a best practice. – Paul Wahl SAP America

Selecting the Right ERP Software

Have you already selected an ERP package? Are you about to embark on an ERP selection initiative? Is your organization already in the early stages of an ERP implementation? If you can answer a definitive 'yes' to any of these questions then you have come to the right place this afternoon.

Our topic today is planning for a successful ERP implementation experience and our first question is when should implementation planning begin? Actually, your ERP implementation began yesterday, even if it has not started yet. The point is simply that ERP implementation planning should begin at the earliest stage of any business re-engineering or information technology strategy that might rely on an ERP solution.

What this really means is that planning for the successful ERP implementation experience begins largely during the definition of your business requirements and the evaluation of alternative software packages that will meet these requirements. If you select the wrong ERP software you will likely experience disappointment, delays and overruns trying to make the wrong software 'right' for your business. That is, if it can be done.

One well known electronics manufacturer has selected no fewer than three ERP software packages for its core business processes over the past decade only to endure the failure of one software vendor, while the rigid business model of a second package caused cancellation of that package's implementation.

This company is now in the midst of implementing the package solution provided by yet another vendor whose product was considered a better fit for what in the end was found to be several unique business models encompassing their varied product lines, instead of a single business model.

It should be well understood that each software package you consider represents an out-of-the-box way of doing business that might, or might not be appropriate in every way to your business. Each package considered will impose its own business logic on a company's strategy, organization and culture. During the evaluation of an ERP system your company must ask itself:

Does this software represent the best way we can do business and is this software the closest fit to how we want to run the business?

Remember that any ERP software you select is the one that you expect will tightly integrate into your business processes and that will require minimal, if any customization. It is really a matter of 'goodness of fit'. Of course that begs the question, will any one package satisfy all of your business requirements? – *The answer is generally no.*

Just as building sophisticated, integrated information systems is not the core, or main business of most companies, companies should not channel any significant effort into making a bad software selection decision work for its core business.

As a general rule, companies should *limit the customization* of their ERP package to the essentials -- functions that are strategically unique or competitively necessary.

Best Practices and ERP Software

For most internal processes the benchmark to pursue is finding ERP software that allows you to do things in the best possible way, not just the right way. So called 'best business practices' are methods that have proven efficient and effective for carrying out necessary activities on a least cost, best results basis.

Acquiring and using best practices knowledge is quickly becoming the norm at most companies today. To achieve this 'best practice state' companies need ERP systems that continually evolve around both accepted and emerging best practice concepts. For instance:

- Evaluated Receipt Settlement
- Vendor Managed Inventory
- Contract Manufacturing
- Activity Based Costing

You will want to specifically consider how well an ERP software vendor implements best practice concepts and what their investment commitment is to keeping its software at the best practices level.

Evaluating ERP Software

Almost all ERP systems available are good at some things, yet not so good at other things. Many of these systems had their start as a specific industry solution. For instance, targeting the specialized needs of steel producers, or automating specific business functions, such as financial processes.

The tendency is for such industry, or functional biases to continue throughout a software product's overall design and thus certain basic or 'original' software functions will be more robust than some of the other 'later generation' functions in the software might be.

As a vendor's ERP package evolves, it will be less likely that it is positioned as an industry specific solution and more frequently positioned as a do-it-all solution. This is where references from your business sector will be important. It won't just be the quality of the references, but also the depth of the references that the prospective software vendor has in your sector that will be important.

It is best to screen out unacceptable software quickly so as to limit your choices very early on in the selection process to no more than five or six ERP systems that you will evaluate on an in-depth basis. A second cut may further reduce this number to only two or three packages.

Use your own business process scenarios if at all possible to evaluate a vendor's software on an in-depth basis. The vendor should minimally demonstrate the following generic business processes to you during your in-depth software package analysis:

- Quote to Cash
- Procure to Pay
- Design to Build
- Plan to Produce
- Manage the Enterprise

Implementation Tools

Consider the tools the software vendor has available for system configuration, personalization, customization and testing. These tools should insure that any system configuration or personalization actions taken would roll-forward into the software's next release.

As for customizing tools, these tools should insure that customizations have the same look and feel as the vendor's core product and that your customizations will tightly integrate with their infrastructure programs and files, including personalization and customization tables, their data dictionary and associated master and transaction data files.

Specific integration tools should exist for introducing external data into the core product's associated master and transaction data files. Such integration tools would complete transactions on an unattended basis, as if they were manually entered into the system and should therefore perform all edits and validations as if the transaction was manually entered. These tools should have acceptable performance levels for both one-time data conversions and for daily processing of transactions from legacy systems that will continue in production alongside the new ERP system.

One tool that may, or may not be valuable is a so called 'implementation assistant', or 'implementation wizard'. These are additional software products (sometimes free) that pose questions about business processes to you. Your replies are in turn used to build values into the underlying configuration tables with information appropriate to your business. Many software vendors and consultants use these tools or possibly pre-built templates of configuration settings to jumpstart an ERP implementation.

However, do not expect entire software configuration to be completed in this manner. Generally speaking, these products are confining and sometimes are not robust enough to completely guide you through the configuration of the software for your business models. They also tend to mask how configuration really occurs at a lower level in the software and that is frequently valuable information, especially when dealing with significant customization, integration or personalization issues.

A final tool to consider is the availability of scripting tools that can assist you in performing testing or in manual data conversions.

System Durability

Any ERP software package you select must be durable. For instance, the average life of most information systems exceeds ten years. When you buy an ERP system, you are buying an infrastructure that should be in place for the long term. The ERP system must scale and grow with you as your business expands. There are many qualifying questions to ask of the vendor that you are considering, including:

- Length of time in business;
- Relative position in the marketplace;
- Are they publicly or privately held;
- Long term business prospects (take a look at their market value trend line, or analyst's comments);
- How have they responded to changes in the marketplace;
- Do they have an open system, operating with different databases;
- Does their system operate on one, or on many different platforms and operating systems;
- What is their platform certification process;
- Is the platform choice (hardware, operating system, database, middleware) you are considering fully certified;
- Number of users in your industry or business sector;

- Number of users (total);
- Percentage of their business done in your primary country of operation;
- Frequency of new releases;
- Percentage of bug fixes versus functionality enhancements within a new release;
- Continuing product technical support options that are available via phone, e-mail, web site or on an on-site basis;
- How are serious software problems escalated and resolved;
- How long has the product been available;
- What is the company's stated direction for the product;
- Are all product features or functions that were demonstrated, available immediately in the software that will be shipped and installed if the deal closes today;
- If certain features or functions will not be available until a later date, when will that be and how critical are those features to your overall implementation.

Must you always buy a 'brand name' software package? To be certain, the five largest players in the ERP marketplace, SAP, Oracle, J. D. Edwards, PeopleSoft and Baan are snagging the lion's share of business. However, there are plenty of other solid players with good products in the second tier, such as Industrial Financial Systems (IFS), Ross Systems, Lawson and Great Plains, just to name a few.

Unfortunately, without a thorough analysis of your requirements and of your expectations as well as the requisite evaluation of the product alternatives that represent a good fit, it would be unfair to reach a premature decision on the suitability of any one product for your specific needs and budget. – *Don't let software vendors tell you otherwise.*

Selecting an Implementation Consultant

Now that you have selected an ERP software package for your company, its time to talk about its implementation. First, it is important to point out that no two software package implementations are the same. Why, because no two companies are themselves the same.

Second, the costs of the implementation itself will likely exceed the cost of the software and any new hardware required by it. As a matter of fact, implementation costs can easily exceed the cost of the software by three to ten times. Therefore, *the selection of an implementation consultant or systems integrator as your implementation partner will be a critical step in the success of your implementation.*

It's all about an approach . . .

Virtually every ERP software vendor and every ERP software implementation consultant will bring an implementation methodology, or approach to the table. These approaches can vary greatly. Some of these approaches are quite exacting – almost painful at times, while another approach may appear fast and loose. Yes, these are two extremes but this is also the reality of the marketplace. So how do you evaluate prospective implementation consultants, including the software vendor?

The obvious first question might be: Why can't I simply 'one stop shop', with the software vendor also serving as my implementation consultant? The answer is that many software vendors have realized they are in the software business, not in the software implementation business.

Quite frankly, these are really distinct businesses. The software factory focuses on building as generic a product as possible while the software implementation consultant focuses on the personalization of this generic product for you. As a matter of fact, even most of the 'big-five' professional service firms have recognized the distinctiveness of the consulting business from their traditional audit and tax lines of business and often organize and run their consulting practices as a separate business.

Several rules of thumb to consider: First, the more personalization you expect to do, the less likely the software vendor will be your best choice, as an implementation consultant will. Many of their methodologies are geared toward 'vanilla' or 'drive-by' implementations of their ERP software. Second, if you have a lot of on-going integration to do with other systems, the less likely it will be that the ERP vendor will be your best choice as the implementation consultant. They are typically not as interested in helping you customize their product, or in using competitive products along side their software. Third, if you expect to do a lot of business process 'stuff' along with your ERP software implementation, again the ERP software vendor may not be your best choice.

Evaluating the Merits of a Consultant's Approach

The first characteristic of a successful ERP implementation approach is one that will orchestrate activity in such a way as to maximize schedule efficiency (i.e. time, budget, and results) while minimizing unnecessary activity along the way. Any approach should emphasize a process orientation, be date driven and clearly identify outcomes, or deliverables.

A key question about an approach is how deliverables prepared in one task or stage will be leveraged or used in subsequent tasks or stages of the ERP implementation. If a clear connection is not present, be sure to question the value of such deliverables. Let the consultant defend their necessity to you.

The second characteristic of a successful ERP implementation approach is it should holistic by definition. How does a consultant implement an ERP system holistically? By integrating people, processes and technology together.

For instance, what specific qualities should you look for in a holistic implementation consultant?

- Ask about relevant software experience, the more ERP packages and platforms the better.
- Ask about business experience – for a change gray hair is good.
- Ask about project management experience – have they been there and done that before and how do they control projects and keep everyone informed and on track?
- Ask about change management experience – how adept are they in introducing new processes, designing and conducting training programs, managing expectations and project scope and in mitigating project risk?
- Ask about knowledge transfer – how will the consultant document their work and insure your company's staff can perform any personalization work after the consultant is gone?

The third characteristic of a successful ERP implementation approach is one that that is adaptable. A good approach is one that can be scaled up, or down as needed. Again, remember that no two companies are themselves the same and therefore no two software package implementations are the same either. A cookie-cutter or purely template-driven approach is simply not sufficient in all cases – it is only a starting point.

The implementation consultant must be adept at aligning their approach to the specific needs of their clients. In short, a good consultant should be flexible in their project approach. After all, part of the reason you are paying consultants in the first place is for their expertise and creativity in designing a 'one off' project for you.

The fourth characteristic of a successful ERP implementation approach is one that consistently manages your expectations. First, let me begin by saying that no ERP

implementation will be perfect. Therefore, the approach must provide for dealing with the unexpected, mitigating risks such as delays, overruns, or changes in scope and minimizing any disruptions to your business. The elements of expectation management include:

- Project management -- how the project is detailed, budgeted and the mechanisms for tracking progress, keeping everyone informed and on track.
- Change management – how your expectations, project scope, functionality gaps, technology problems and other project risks are identified, prioritized, tracked and resolved.
- Quality assurance – how the implementation partner provides internal quality assurance or client satisfaction. The process should provide you with reasonable assurance that a consultant's work is subject to some degree of peer review and that processes exist for measuring your ongoing satisfaction with the consultant's work.

Must the consultant you choose be a business partner of the software vendor? There are many good consulting firms that are not necessarily business partners with the ERP vendor you have selected. That does not necessarily disqualify them. As a systems integrator they frequently complement their strengths through partnerships with other consultants. As a matter of fact, during software selection there may actually be some advantage to using a firm that does not have any formal partnerships with the ERP software vendors you're evaluating – It prevents the possibility of any conflicts of interest.

Note that the smaller the selected ERP software vendor is, the less likely you'll find as many implementation consulting choices. If you are taking a best of breed approach that encompasses integrating software from multiple vendors, you will also have fewer consulting choices.

The consulting firms that you invite to bid for your implementation work should identify for you what pieces of the work they will personally do versus what pieces of the work they will sub-contract, or consider outside the scope of their bid. The more complicated and challenging your implementation is, the more likely the consultant will use other specialists and therefore serves in a 'systems integrator' capacity rather than as simply an implementation consultant.

Evaluate all firms equally. Remember that consultants are in the personal service business. Select an implementation consultant who has an approach you're comfortable with, is demonstrably qualified to do the work and lastly, is an organization with whom you are comfortable doing business with and who prices their work fairly.

Very few software vendors or implementation consultants will provide you with bad references. Typically they want to put their best foot forward with prospects. So you need to query your network in addition to checking out their references. If by chance

you do receive a sour reference, follow up on it. Sometimes these things may be old news or an isolated case and really have little relevance to your evaluation.

Qualify your consultants, not just the firm, but the consultants themselves. Generally speaking, most consulting firms will make their pitch to you with the consultants who will actually perform the project work. If not, be suspect that you're being pitched by the 'A' team and that the project work will be completed by the 'B' team.

Whether it's the 'A' team, or the 'B' team that will do the work, make sure that your internal team members interview the consultants as prospective members of your overall project team. Make sure you are comfortable with them. Most consulting firms will allow you to pick the team. After all, you are the one who is paying the bill. Finally, be aware that consulting is a very transient business (it is no different than the revolving door that possibly exists within your own information technology department). Therefore, be prepared for the inevitable: The longer your ERP implementation project is in duration, the greater the possibility is that you'll lose one of your handpicked consultants along the way.

Business Process Re-Engineering and ERP Implementation

When I hear a client immediately suggest that ‘what we need is a new system!’ as a consultant, I begin thinking, now where have I heard that before? So often systems are both blamed for and suggested as the remedy for the ills of simply bad business processes.

To be fair, business process re-engineering also has a bad rap. It’s been viewed all too simplistically as a way to downsize the organization, an excuse for cutting jobs and management levels. However, business process re-engineering is really aimed at the elimination of non-value adding work and on leveraging the capacity, or throughput of business processes and business resources.

At first glance a new system, especially an ERP system, will potentially solve many issues a company may have. However, upon careful analysis, if a company has significant process issues, a new system alone may not solve any of its problems. In the words of an old and dear friend – *‘it may simply represent a new, or different way to do exactly what was done before’*. – *W. C. Walker, Viskase Corporation*.

There are two schools of thought on business process re-engineering and its impact on ERP system implementation. The first approach is to engage in a more radical, or ‘from the ground up’ re-engineering effort. This approach suggests that before you can automate any process, you must seek to first understand that process and to then simplify that process. For instance you might ask of your processes:

- Should we be doing this task at all?
- How can we do this task better?
- Are there accepted best practices on how this task (or process) should be done?
- What are our benchmarks for the completion of this process?

If you have selected a best practice based ERP software package, you may have all but eliminated the need for a radical business process re-engineering project – why? Since the ERP package is already best practice based, a significant re-engineering of the company’s business processes should be unnecessary. In this case business process re-engineering would be evolutionary, not revolutionary.

During the implementation of the ERP software package your business processes are modified in such a way that the new system can be used appropriately in your business. It’s really about the alignment of your business processes with the new ERP system’s capabilities. This softer form of business process re-engineering will represent a natural evolution, or revision of your business processes toward the package-provided best business practices.

In this latter re-engineering approach, before the business process can be automated through the use of the ERP software package, you need to:

- Understand the existing process;
- Understand the best practice based process in your ERP software package;
- Synthesize your existing process into the best practice based process in your ERP software package;
- Identify any functionality gaps and assess the business impact such gaps present.

Synthesizing your existing processes is the tough part. First, you'll need to re-evaluate steps in your processes if they are not present in the best practice based process in your ERP software package. For instance, are they truly value-adding activities, or were they present in the process to make up for shortcomings in previous information systems? Can you adjust business policies and procedures or is that simply not possible? In short, strive for process simplification. *Having the right people in place from your business functions is a critical success factor to this synthesizing process.* (Later on we'll cover the prerequisites for team member selection.)

Yes, radical business process improvements are possible simply by implementing an ERP system. However, using an ERP system is often of greatest benefit to a heavily process-challenged, manually intensive organization. *In a company already well along in its' business process rationalization efforts, using an ERP system's best practice based processes will likely yield only incremental, not radical business process improvements.*

The Softer Side of an ERP Implementation

Although your boss will likely ask ‘What are these line items for?’ or ‘Is all of this necessary?’ you will soon realize that these items are some of the most important line items in your project budget. They are project management, training and change management. The tendency is to underestimate both the cost and value of these line items to a successful ERP implementation. More analysis regarding the importance of these factors to your project’s success will follow in later sections.

For now however, our focus will be on what I refer to as the ‘soft’ factors that figure heavily into the success of an ERP software implementation project. They include:

- Project Sponsorship
- Project Team Member Selection
- Project Team Empowerment
- Ongoing Communications about the Project
- Managing Your Consultants

Project sponsorship – it does not end with simply an authorization for expenditure. That is only the beginning of a longer journey. Project sponsorship should occur at the highest levels in the organization. The project steering committee should be comprised of a key executive from each major business function, the project team manager and their consulting team counterpart.

The selection of the project team manager and the project working team itself should not rely on, nor tap those who are simply ‘available’. Rather, the team should be comprised of experienced, highly regarded business experts who can effectively leverage your ERP software investment. Also remember that an ERP implementation project is not primarily an information technology initiative; it should be viewed as first and foremost, a business initiative. Be sure to select a project leader who can appreciate and share such a project vision.

Remember too that these individuals will likely devote all or most of their time to this project and likely will work above and beyond the call of duty for the duration of this project assignment. These individuals should be compensated on an ongoing basis for their dedication and effort to this project and through an end of project stipend.

Won’t we lose our project team to a consulting firm anyway? Not necessarily. First of all, ERP related hiring has slowed dramatically as the Y2K blip in ERP implementations is all but over. Consulting salaries have also leveled off and overall experience requirements are up. Second, not everyone wants to be consultant. Those who enter the ranks of consulting face frequent travel, generally long hours and will need to enjoy working without a desk, office or even a cubicle to call their own. So, if you treat your project team members with dignity, along with the financial rewards suggested, they will likely stay with you. And after all, since these individuals represented your best and

brightest, when the project is successfully concluded it may be the time to recognize them with greater responsibilities.

I can not say enough about the necessity of project team empowerment. The future of your business processes rests in the hands of your project team. If you have selected your project team wisely, their expertise and guidance in the usage of the software should be definitive. As to the details of their work, the project team should be largely self managed and empowered.

Periodic project communications prevents rumors and helps to insure everyone is in tune with the expectations of executive management. Under the president's signature, an initial letter should be addressed to every employee indicating that ERP software is being evaluated for the company, what the project's vision is and when a selection is expected to be finalized. When the selection is finalized, a similar letter should be distributed, including a rough project timeline. Each letter should specifically identify the project team, what the project team will be doing and finally, encouraging everyone's active cooperation and support in helping the project team to complete their work.

Periodically thereafter, an executive briefing letter should be prepared and distributed as the project progresses, usually as a major phase in the project is completed. I have found that people really want to know what is happening and why, how it will affect them and when. Such communications also enhance the accountability of the team to their deadlines; they will not want to be associated with failure, and will therefore generally not provide unrealistic dates or make otherwise unrealistic projections about their work.

The final element is the ongoing necessity to effectively manage your consulting relationships. I have found that consultants fall into three categories:

- Consultants who want to leave before the job is done.
- Consultants who don't know when it's time to leave.
- Consultants, who know when it is time to move on and work with you to prepare an exit strategy from the project.

The latter alternative is of course the desirable state of affairs. Be assured that your consultants will offer to provide as many services as you can imagine for as long as you can tolerate them. Remember the role you had in mind for the consultants from the start and make sure that your expectations about their engagement scope remains in check.

There is a serious disease that afflicts many ERP implementations. It's called scope creep and it happens frequently. An important role should be for the consultant to manage your expectations about what the software will do and to what extent you'll entertain changes to the software. However, this does not mean that they will keep scope creep in check. That will rest with you. Remember that scope creep can represent additional revenue streams to the consultant and conversely, cost overruns to you. Because of this potential conflict of interest, the consultant should present to you, for your express approval, any changes in the scope or duration of their work.

It is your responsibility to insure that the consultant is managing your expectations effectively, is serving in your best interests (e.g. preventing large-scale customization to the core functionality in the ERP package, keeping the project on a strict timeline and budget) and that the work they are undertaking is value-adding.

Project Scope

What about the so-called ‘big bang’ versus ‘phased roll-out’ approach in ERP implementation? The selection of the ‘right way’ to roll out your ERP software is quite circumstantial. If you have a limited number of locations, a ‘big bang’ approach is very attractive. If you have numerous locations, especially across borders, ‘phased roll-outs’ are more appropriate.

Another common approach is to begin by rolling out three lower risk, well-defined financial applications that are frequently single location business functions: general ledger accounting and financial reporting, non-inventory accounts payable and fixed assets – a so called limited scope roll-out. Generally speaking, these modules from an integrated software package will operate on a standalone basis. In addition, the underlying organizational structure and chart of accounts needed here also represent the foundation for integration of the vendor’s other software modules into the accounting system.

In my experience I have seen all three approaches work under the appropriate circumstances.

Project Risks

Most importantly when considering how to plan your ERP software roll out is the consideration of project risks. Project risks must be mitigated. The appropriate rollout plan, even if more costly or time-consuming should carefully balance the impact of any significant business interruption against the increase in implementation costs. But there are tradeoffs. Long term or multi-year ERP implementations can develop cases of significant scope creep, cost duplications, lost momentum and disappointing returns on investment.

Stepping Through an Implementation Approach

The remainder of this paper focuses on stepping through a typical ERP package implementation approach. The approach illustrated here is best practice based, representing my cumulative knowledge and experience drawn from multiple ERP system implementations. The implementation approach is the basis for building and executing a successful project plan.

This approach serves as a framework for the design and development of a project plan specifically tailored to achieve the implementation of any ERP system, breaking the enterprise system implementation into stages that logically divide a complex project into plannable units of work. These stages work together to ensure that the implementation effort is planned controlled and monitored. Within each stage of this approach are specific activities, or tasks, and there are deliverables, or outcomes associated with each such task or activity.

As you evaluate software vendors, implementation consultants and solution integrators, you should encounter similarities to the approach illustrated here. In my experience, I have found a successful ERP implementation approach should include the following stages:

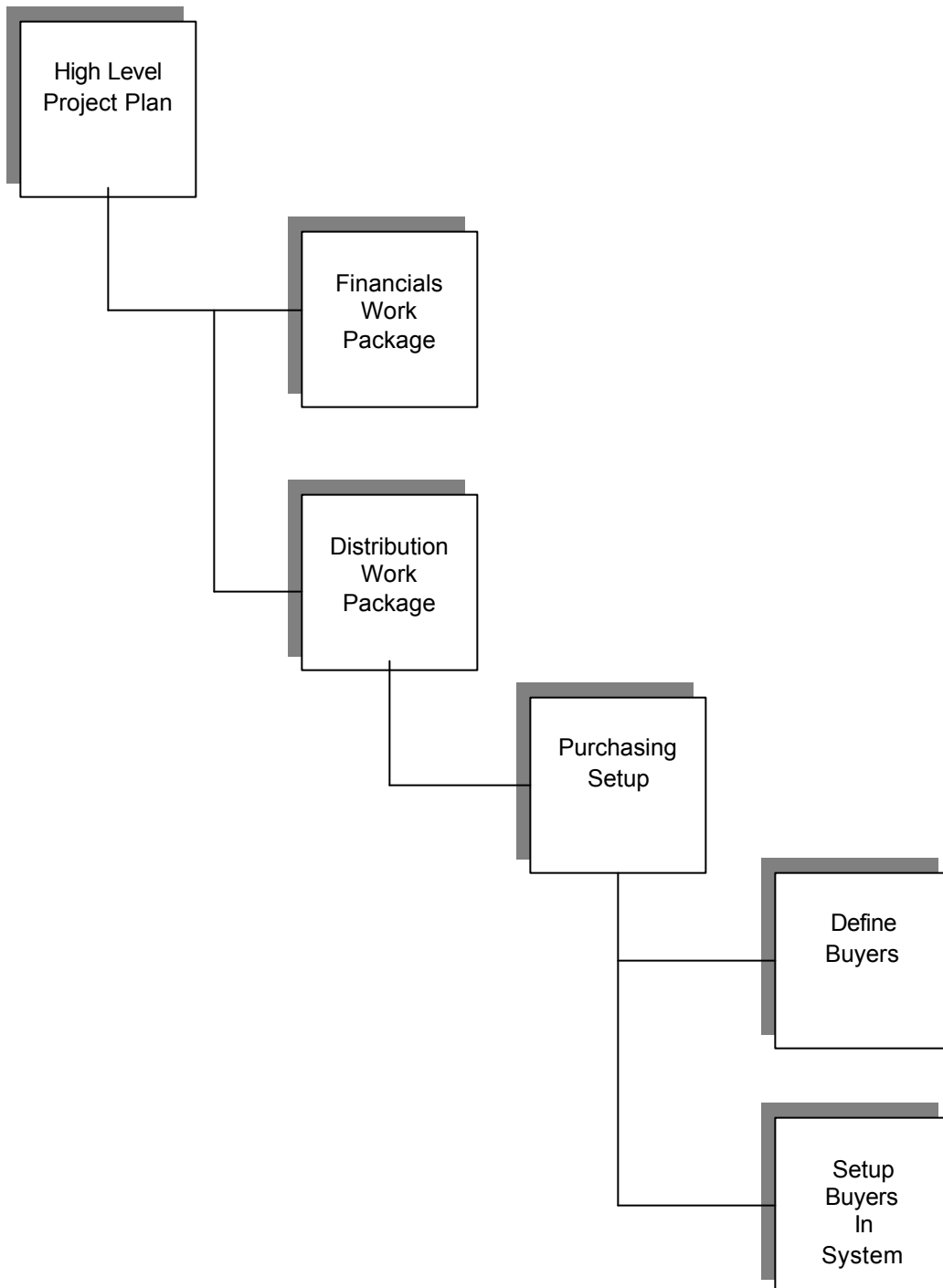
- Charter the Project
- Select the Project Team
- Educate the Project Team
- Model the Business
- Configure the Software
- Prototype Business Processes
- Identify Functionality Gaps
- Develop Technical Solutions
- Document Configurations, Customizations, Personalizations and Processes
- Establish Production Readiness
- Train End Users
- Go Live
- Establish Steady State
- Perform Periodic Health Checks
- Complete Periodic Software Updates

It must be clearly pointed out that a lower-level, or detailed work breakdown structure, or work package must be developed for each stage of your ERP implementation project.

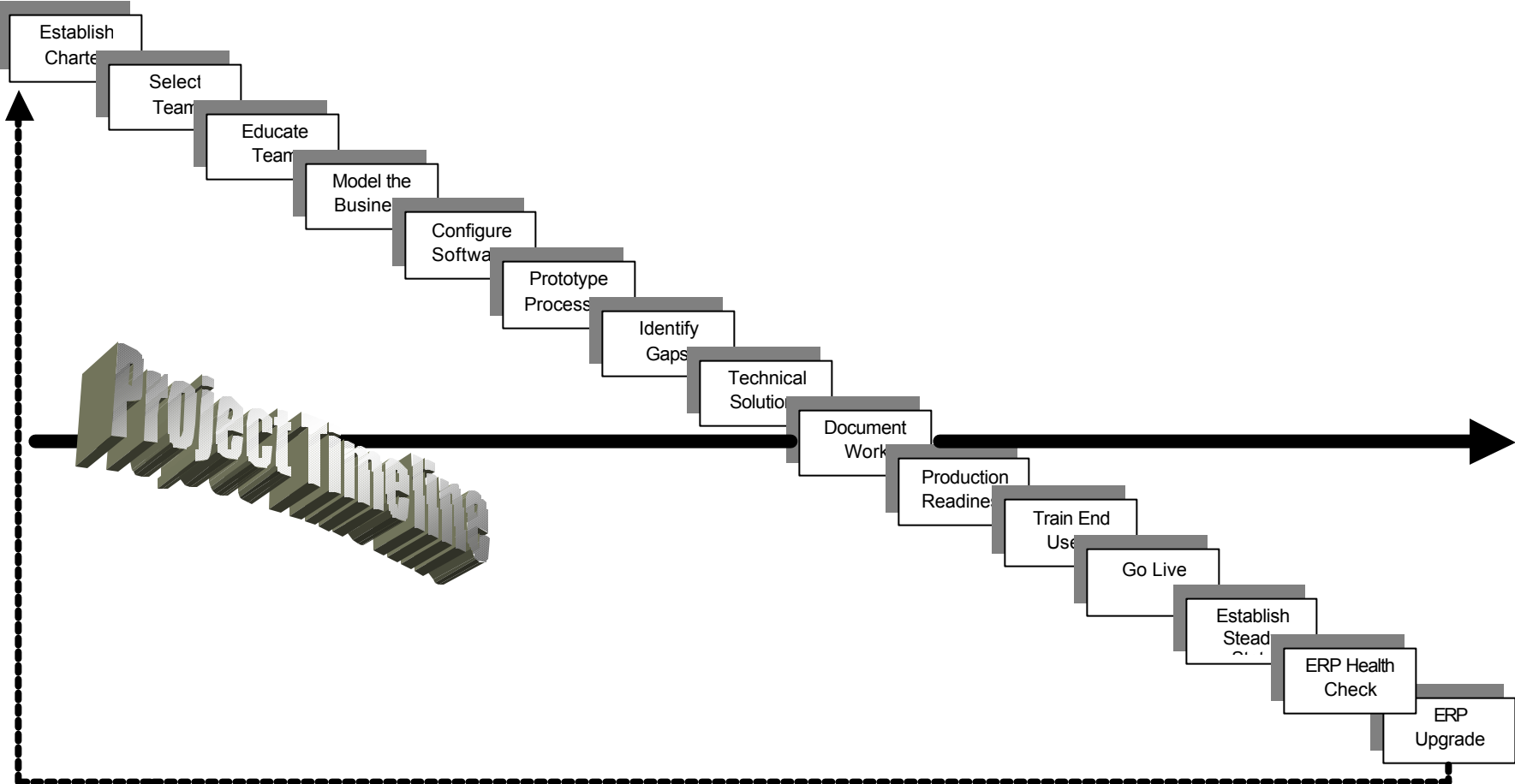
For instance, within the work package you would identify specific configuration steps. You would have a high-level master project plan for all of the stages of your project and multiple sub-projects. For instance, one covering each major functional area or possibly

for each location being implemented. It is best to use a robust computer-based project-planning tool such as Microsoft Project or Primavera Systems Primavera Project Planner.

A Typical ERP Implementation Work Breakdown Structure



A Best Practice Based ERP Software Implementation Approach



Charter the Project

A clear understanding of the ERP system implementation strategy will be developed. Specifically, a detailed implementation plan and project timetable will be developed of what needs to be accomplished and who is going to accomplish it. Project working and executive teams will be defined and staffed. The project scope and objectives will be defined.

Select the Project Team

Formal project executive sponsorship is established. A working team of the best and brightest is assembled from your representative business areas. A leader is appointed for this working project team. Consulting participants are interviewed and added to the team.

- *The Project Executive Team*

The project executive team, or project steering committee will meet to review progress, assess project risks and decide on project scope, budget or schedule changes. The basic agenda for these meetings will be to review the overall project status using the project plan, the project budget and the project issues list as guiding documents.

- *The Project Working Team*

The project working team will meet to review progress, determine project status, identify issues and tasks, identify project risks and project scope changes. The basic agenda for these meetings will be to review the project plan, individual action plans and task status, project issues and to propose resolutions to issues.

Educate the Project Team

The project team receives overview training in project and change management, business process analysis and modeling techniques, and specialized education in the base ERP software functionality. The base ERP software environment is defined and the base delivered ERP software is installed into this environment. Any necessary project team hardware is installed and made operational.

Model the Business

An analysis of current business practices within all affected functional areas will be completed. During this stage of the project, a full understanding of the current business environment, business practices and processes as well as an understanding of the roles and responsibilities of each member of the software user community will be developed. Best practices contained in the base ERP software product that you will use in your company are identified.

There are many ways to model your business. I have found that a combination of visual and written process details works best. For instance, I have created easy to use tabular style business process analysis document templates using as Microsoft Word. As for a visual business process-modeling tool, my preference is allCLEAR, from SPSS Incorporated.

With allCLEAR you do not need to be an artist or draftsman, nor do you need to spend countless hours aligning flowcharting elements and lines found in many other similar products. The tool can literally create perfect charts for you by simply typing in short sentences.

A key deliverable of this stage will be a series of transaction processing scenarios. These scenarios will be the basis for configuring and prototyping the software's business functionality and identifying any potential software gaps.

Configure the Software

In this stage, a solid business and technology architecture is sought. Define how the base ERP system functionality will be used. The base delivered ERP software product is configured around the business transaction scenarios, the business environment and the user community. It is important to note that tailoring, or personalizing the base or core ERP software, does not include the modification or customization of any of the source code of the base delivered ERP software product. *A key tenet of a successful ERP implementation is to avoid customization of the core product where and when possible.*

Key deliverables from this stage include revised transaction scenarios to be used during later project stages. The business environment, workflow and user community definitions are revised as needed during this stage.

Prototype Business Processes

Proof of concept occurs in this stage. Transaction scenarios are tested. A complete understanding of how the base ERP software works is sought and these processes are documented. Any software gaps are identified for further action. The following generic business processes are typically exercised during this stage:

- Quote to Cash
- Procure to Pay
- Design to Build
- Plan to Produce
- Manage the Enterprise

As prototyping or 'proof of concept' occurs, testing and acceptance criteria are developed.

Identify Functionality Gaps

Further testing of any previously identified transactions containing gaps occurs. A full understanding of software gaps is sought and workarounds are devised and modeled. Custom transactions are designed to replace, or extend the functionality of the base delivered ERP software product.

Develop Technical Solutions

Custom transactions are developed against the previous specifications to replace, or extend the functionality of the base delivered ERP software product.

These custom transactions typically represent modifications to the source code of the base delivered software product. Custom transactions are completed and tested, then integrated into the overall workflow process of the ERP system.

A data migration or conversion plan is devised and executed. Master data and transaction-related data from existing, legacy systems are identified and analyzed. Tools or programs are used to scrub, edit and load this data into the appropriate ERP system data files or data tables.

Document Configurations, Customizations, Personalizations and Processes

User procedures, documentation and training are essential parts of the end-user support infrastructure that must be in place when the ERP system moves into production and replaces the current infrastructure and processes.

This stage will commence as workflow and functionality issues are finalized. Business policies and procedures are revised and rewritten as needed to support the new ERP system. A training plan is devised and executed. The design and construction of training classes and classroom collateral materials is completed.

A training schedule is devised; participants are identified and scheduled. A training environment is defined and training data for hands-on exercises is constructed and created in this environment. Any necessary training facilities and hardware is obtained and made operational. Train the trainer training is provided. Trainers practice their timing and presentation skills. Trainers must insure they have complete competency in the software functions and business processes they will be expected to train.

Establish Production Readiness

Verification of the actual stored data and the validation of business process audit trails contained in the data is an important step in assuring the accuracy and consistency of converted data.

Testing plans should allow for individual component, module, and interface testing and finally end-to-end testing of the entire ERP system. This includes tests of converted data

that your go-forward processes rely on. All of the detailed work during prototyping and production readiness makes 'going live' a decision point rather than an additional task to fulfill at the end of a project.

Train End Users

In this stage primary attention is given to the delivery of end user training. Training should be conducted as near as possible to the go-live date. Hands-on training should be provided. Walk participants through business processes as a part of the training.

Provide participants with exercises built on sample or demonstration data drawn from data, which they are accustomed to. Provide ample time to practice during formal training classes and a sandbox environment where participants can continue to practice and improve their competency with the new ERP software until and after the 'go live' date.

Use a classroom whiteboard as a parking lot for unresolved questions, problems or issues that arise. Use breaks between classes to address and resolve these issues. Review the previous days' topics and parking lot issues before moving on. If needed, provide written follow up of all parking lot issues resolved after training ends.

Go Live

A readiness assessment is completed. The management and working teams assess the overall readiness to 'go live' with the software. A 'go live' decision is made and the software launch countdown begins.

The loading of master, configuration, historical and transactional data into the production environment is performed.

Your company starts conducting business through the new system. This may, or may not be done, in a parallel mode with dual entry of business transactions into both the new ERP system, and into any legacy systems.

Establish Steady State

Support begins coincident with live production. During this stage, problem resolution and change management procedures are critical.

System performance monitoring should be conducted to insure that both online and batch processing standards are met. The amount of tuning required will depend on initial results in production. There may also be performance issues encountered that require changing processes or ERP system configurations.

At this stage a smooth and orderly transfer of knowledge to your personnel should occur and your implementation consultants should be rolling off the project.

Perform Periodic Health Checks

Keeping your ERP system healthy is an ongoing process rather than a stage in the project. It begins after the first component, or module of an ERP system enters production and extends throughout the installed life of the ERP system.

Periodic health checks should assess both the functional and technical results of the system. You will constantly want to address whether the ERP system is achieving its original objectives and whether its full benefits are being realized.

Complete Periodic Software Updates

As mentioned previously, periodic health checks should assess the functional and technical performance the ERP system. Active involvement in user groups and upgrade information programs or subscription services provided by the vendor will allow you to periodically assess whether an upgrade is desirable.

Upgrades generally provide cumulative fixes to known software bugs and often provide performance enhancements through design changes that leverage system performance through other middleware, operating system and database infrastructure upgrades.

The software factory should also be making functionality enhancements that may further extend the value of your ERP system, or possibly eliminate the need for a remaining legacy system.

Upgrades can pose a significant challenge on an ongoing basis if you have made extensive customizations or have significant integration points with other legacy systems.

As a general rule, for an established and stable ERP software package, the vendor should limit upgrades to an annual frequency.

Don't upgrade immediately when a new release is available. Wait about six months, while you let others shake out any problems.

Try to stay no more than one release level behind the current version of the software. There are several reasons for following this recommendation:

- Straying too far from the pack lessens the quality of vendor support you'll likely receive.
- Most upgrades must be sequentially applied. This may require that if you fall behind several release levels that you will need to 'catch up' before applying the latest release.

You can be bombarded with lesser releases that need to be applied against your base software. These are known as interim releases, bridge releases, quick fixes, patches, service packs or temporary fixes. These releases are used to resolve specific problems. The problems they address will span from the mundane to the severe. Carefully weigh

the merits of each such fix, or patch to your system and what ramifications it may have on your system, especially if you have invested in substantial customization, integration or personalization of the core, or base ERP system.

Another release category may be the year end release. These are generally available for specific regulatory changes (i.e. new 1099 or W2 information reporting formats) and most vendors will provide these for all supported users of their software.

Is an ERP Implementation Ever Really Complete?

The astute reader already knows the answer to this question. No, a successful ERP implementation is never really complete. Why? Recall from the previous section my comments that keeping your ERP system healthy is really an ongoing process, rather than a stage in the project. The successful ERP implementation project is therefore a continuous closed loop system, with the ERP health check providing the impetus for continuous improvement and tuning.

Periodic health checks should assess both the functional and technical results of the system. You will constantly want to address if the ERP system is achieving its original objectives and whether its full benefits are being realized. *Tuning and upgrading of the ERP system to achieve higher performance levels and greater business value is a journey, not a destination.*

Of course there is another way to answer this question: Your business is not static, it is dynamic. Therefore, by definition your business processes and infrastructure, which includes your ERP system, are constantly changing to reflect the dynamics of your business.

Some Thoughts about Technological Infrastructure

During the content review phase of my preparations for this presentation I received some excellent feedback regarding a topic that I admit to having treated only in passing previously. That topic is one of technological infrastructure. Although the primary purpose of this paper is to review an approach for achieving a successful alignment of the business into an ERP infrastructure, it is not done without adequately planning and implementing a robust technological infrastructure that allows your ERP software to run successfully.

Most businesspeople do not grasp technology well. Even manufacturing managers, engineers and accountants – all generally technically astute and detail oriented individuals sometimes fail to appreciate the enormous amount of behind the scenes complexity of computer software. As computer software goes, ERP systems represent some of the most sophisticated amount of coding, or programming anyone has ever done on a computer. A typical ERP system has thousands of computer programs and millions of lines of instructions. Hundreds of thousands of hours have gone into the design and development of the software.

ERP systems are large, cumbersome and resource intensive software products. They have to be. Why is that so? An ERP system is expected to work for just about every condition and in just about any type of business. The ERP software vendor must simultaneously provide for both granular and generic capabilities in their product. Of course there is also the expectation that ERP software will be as simple to configure and use as an everyday product might be, for instance such as using a microwave oven or a toaster.

For some of you a quick reaction might be to select a new ERP software package based solely on the type of technology infrastructure you have in place. It should not be. Remember that an ERP software decision should be a primarily a business driven initiative, not a technology driven initiative.

As more and more ERP software vendors move toward an open systems software architecture, the less impact hardware cost, availability or your organization's previous infrastructure experience has in the ERP software selection equation. In short the technological infrastructure behind an ERP system has become largely a commodity purchase but the importance of technology to your overall decision does not diminish.

A good general rule I have found is to always buy or lease the fastest and largest computer available at the time of your decision. Also note that you will always need more disk space than you can imagine ever needing. ERP systems create and store a massive amount of data. These systems also slow down as their databases grow.

What is an open systems architecture or model as it relates to ERP? Generally speaking an open system model in the ERP world means that the ERP software itself operates with

multiple hardware, operating system and database system combinations. An open system should not be confused with client server computing although an open system is generally a prerequisite for client server computing.

For instance, many older generation ERP systems required a specific hardware, operating system and database system combination such as an IBM midrange, which only used an IBM's propriety operating system and included a built-in database. That old model is gone now. Today you must select from literally a smorgasbord of operating system, hardware and database choices as an ERP system infrastructure. In addition there is the client-server-computing model, which introduces another element of complexity referred to the process as middleware. Middleware is best seen as a computer network's traffic cop. In the client server model a network of computers are brought together. Middleware is then used to distribute workload and data across a network of interconnected computers, often in multiple locations.

A popular addition or complement to an ERP system, especially if you perform a substantial amount of analysis on your data, is a data warehouse. A data warehouse provides a storage and retrieval system that can be used to archive and summarize transactions from your ERP system. Some ERP systems are now building these kinds of products into the ERP system itself under the category of a business information or business intelligence system. Such a complementary data warehouse is also a major resource consideration that will impact your overall technological infrastructure design.

I wish I could say that technology decisions have become easier given that ERP software vendors have begun embracing an open systems infrastructure – that it could simply be a plug in and go process. On the contrary, it is not that way. Getting the right technological infrastructure in place is a difficult and complex decision. Getting this decision wrong can lead to a major setback in the success of your ERP implementation. I have seen this happen more than once. Some for instances include:

- An improperly tuned machine that caused interactive processes to accept and process a new screen of information every 20 seconds when sub-second response time was needed;
- An undersized machine causing with insufficient computing power to run batch processes within the after hours time window available,
- A database and machine combination that was not vendor certified,
- Lack of disk space needed to create multiple environments to support configuration, training, testing and production activities simultaneously.

Early on I mentioned the importance of insuring that the ERP system you select is certified in the hardware environment you have, or will evolve to. Throughout this discussion you should now be in a position to appreciate that importance of architecting a robust technological infrastructure for your ERP software.

The technological infrastructure represents an entirely separate initiative that must parallel the ERP decision. The right technological architecture must be chosen and installed and it is critical to the success of your ERP software implementation.

A Final Thought on Approaches

This paper has focused on stepping through an ERP software package implementation approach. The approach illustrated is best practice based, representing my cumulative first-hand knowledge and experience drawn from multiple ERP system implementations and my continuing research of multiple ERP system implementations.

Recall that a key point is that the implementation approach is the basis for building and executing a successful project plan. Much up-front and on-going work is required to fully define and build out lower-level work plans. However, on an on-going basis, project management, change management and expectations management are integral elements in a successful ERP implementation experience.

Once you have the plan, you must work the plan; *there are no substitutes available for the enormous amount of effort that is necessary in implementing an ERP system*. ERP Buyers can't do 'it' without implementation partners. Implementation partners can't do 'it' without real resources from your functional business units.

Is one ERP implementation approach better than another? It has been my experience that one approach is not really any better than another approach. However, there are other factors that influence project success. Those factors have been generously covered throughout this paper. When it comes to a successful ERP implementation experience, you will do well to carefully follow the advice of two well-known general management sages:

Look Ahead, Plan Ahead, Move Ahead – Karl Albrecht

If you can't measure it, you can't manage it – Peter Drucker

An ERP Bibliography

The sources and references that I have relied on include:

Enterprise-Wide Software Solutions, Integration Strategies and Practices, Sergio Lozinsky, Addison-Wesley, ? 1998 by Sergio Lozinsky.

On Time Technology Implementation, How to Achieve Implementation Success with Limited Time and Resources, Bennet P. Lientz and Kathryn P. Rea, Academic Press, ? 2000 by Academic Press.

Putting the Enterprise into the Enterprise System, Thomas H. Davenport, Harvard Business Review, July - August 1998, ? 1998 by Harvard University.

The USA Principle, A three-step solution to reengineering, Karl M. Kapp, Manufacturing Systems, August 1996, ? 1996 by Manufacturing Systems.

The USA Principle: The Key to ERP Implementation Success, Karl M. Kapp, APICS: The Performance Advantage, June 1997, ? 1997 by APICS.

The Centerless Corporation, A new model for transforming your organization for growth and prosperity, Bruce A. Pasternack and Albert J. Viscio, Fireside Press, ? 1998 by Bruce A. Pasternack and Albert J. Viscio.

Simplicity, The New Competitive Advantage in a World of More, Better, Faster, Bill Jensen, Perseus Books, ? 2000 by William D. Jensen.

An ERP Vocabulary

This brief guide provides a translation of many ERP software related terms that you will encounter in conversations with software vendors and implementation consultants.

Assemble-to-order. A make-to-order product for which key components used in the assembly or finishing process are planned and stocked in anticipation of a customer order. Receipt of an order initiates assembly of the finished product. This is useful when a large number of finished products can be assembled from common components.

Base functionality. *See core functionality.*

Base system. *See core system.*

Bucketed system. An MRP, DRP, or other time-phased system in which all time-phased data is accumulated into time periods or "buckets." If the period of accumulation is a week, then the system is said to have weekly buckets.

Bucket-less system. An MRP, DRP, or other time-phased system in which all time-phased data is processed, stored, and displayed using dated records rather than pre-defined time periods or "buckets."

Capacity requirements planning (CRP). The function of establishing, measuring, and adjusting limits, or levels of capacity. It is the process of determining how much labor and machine resources are required to accomplish the production workload within a given time span. Open work orders and planned orders in the MRP system are considered inputs to CRP, which "translates" these orders into hours of work, or demands against available labor and machine capacity.

CMMS. A computerized maintenance management system. A system to provide for the planning and management of scheduled or routine preventive as well as restorative or unscheduled maintenance events. These capabilities are often integrated into an ERP system.

Conference Room Pilot. The process of modeling or prototyping a business process using the ERP software functionality after it has been configured against your business requirements. Piloting of processes may be iterative until proof of concept is reached. Pilots are often conducted for specific software modules, and then as a total process; or an integrated conference room pilot. Synonymous with *application, or integration testing and acceptance.*

Configuration. *See personalization.*

Conversion. The conversion of data from existing systems into an ERP system. *See also migration.*

Closed-loop MRP. A system built around material planning that includes the additional planning functions of sales and operations (production planning, master production scheduling, and capacity requirements planning). Once this planning phase is complete and the plans have been accepted as realistic and attainable, the execution functions come into play. These include the manufacturing control functions of input-output (capacity) measurement, detailed scheduling and dispatching, as well as anticipated delay reports from both the plant and supplier. The term "closed loop" implies that not only is each of these elements included in the overall system, but also that feedback is provided by the execution functions so that the planning can be kept valid at all times.

Core functionality. The unmodified standard or base functionality provided by a software package. The core functionality in a software package can frequently be tailored, or configured or personalized without requiring modifications to the software programs, or source code itself.

Core system. The central and foundational systems of an ERP system.

CRM. Customer relationship management, as in a customer relationship management system.

Customization. When the unmodified standard, or base functionality provided by a software package is not found to be sufficiently robust enough to support a business process the program(s) that provide such functionality must be modified, or customized to fulfill those business process or functionality requirements.

Design to build process. Essentially the new product development and engineering steps required for bringing a product idea from concept to a production ready state. Synonymous with engineer-to-order.

Discrete manufacturing. Production of distinct items such as automobiles, appliances, or computers.

DRP. Distribution Requirements Planning. In a company that serves as a distributor, not a manufacturer, DRP is used. It is a variation on MRP. *See also MRP (materials requirements planning.)*

Engineer-to-order. Products whose customer specifications require unique engineering design, or significant customization. Each customer order results in a unique set of part numbers, bills of material, and routings on the plant floor.

ERP. Enterprise Resource Planning. Conceptually, it is the planning, tracking and managing of all resources utilized by the business in fulfilling its business activities (i.e. making a product, performing a service).

ERM. Enterprise Resource Management. *See ERP.*

Functionality Gap. A functionality gap occurs when the unmodified standard, or base functionality provided by a software package is not found to sufficiently robust enough to support a business process without the modification of the program(s), or source code related to that business process.

Gap. *See functionality gap.*

Integration. The process of extracting information, or data, from one system and introducing that same information, or data, to another system. The desirable state of affairs is the *seamless and unattended* integration of data from one system to another. The ERP system does this naturally. A legacy system or a best of breed external add-on, or bolt-on software product may not do this naturally. The ERP system may provide for external integration of foreign software through entry points for foreign data into the ERP system, or through exit points in their software, or programs to interface with a legacy system, custom program or add-on or bolt-on systems.

Interface. *See integration.*

Legacy system. An existing system that is being replaced by, or is supplemental to an ERP system.

Manage the enterprise. A generic reference to business processes that do not directly impact product flows from suppliers, or to customers. These processes include business performance measurement, resource administration and maintenance activities, such as plant maintenance, fixed asset accounting, general accounting and human resources.

Make-to-stock product. A product that is shipped from finished goods, "off-the-shelf," and therefore is finished prior to a customer order arriving. The master scheduling and final assembly scheduling are conducted at the finished goods level.

Migration. The process of transforming, or converting data from an existing system into an ERP system.

Mixed mode manufacturing. A manufacturing process may combine elements of several manufacturing models. For instance, a plastic bag manufacturer may produce coils, or rolls of plastic tubing using process (batch) manufacturing techniques. As customer orders are received, the bag manufacturer assembles, by cutting and sealing the tubing on one edge, to form plastic bags to the dimensions indicated on the customer order.

MRP II. Manufacturing Resource Planning. A method for the effective planning of all resources of a manufacturing company. Ideally, it addresses operational planning in units, financial planning in dollars, and has a simulation capability to answer "what if" questions. It is made up of a variety of functions, each linked together: business planning, sales and operations (production planning), master production scheduling, material

requirements planning, capacity requirements planning, and the execution support systems for capacity and material. Output from these systems is integrated with financial reports such as the business plan, purchase commitment report, shipping budget, inventory projections in dollars, etc. Manufacturing resource planning is a direct outgrowth and extension of closed-loop MRP.

MRP. Material Requirements Planning. A method for the effective planning of the *material* resources of a manufacturing company. *See also* *DRP and manufacturing resource planning (MRP II)*.

Order to cash process. *See quote to cash process.*

Quote to cash process. The basic business process of taking an order from a customer and fulfilling that customer order. Includes the making, or procuring of materials, delivery of the material to fulfill the order, billing the customer, customer payment receipt and the application of the customer payment against their account.

Personalization. The process of adapting, configuring or tailoring the unmodified standard, or base software functionality. This capability is generally provided through a series of switch settings and table settings or values that represent your standard business rules. These settings subsequently influence the software package's program(s) to function in such a way that represents your specific business process, transaction or event.

Plan to produce process. The combination of business processes leading from a sales forecast, or firm order, through raw materials procurement and transformation through the various manufacturing stages until completion of a finished product.

Process manufacturing. Production that adds value by mixing, separating, forming, and/or performing chemical reactions. It may be done in either batch, or continuous mode.

Procure to pay process. The basic business process of placing an order with a supplier and receiving the goods or services associated with that purchase order. Includes the procuring of materials, receipt of the material to fulfill the order, logging of a supplier's request for payment, supplier payment and application of payment against their account.

Proof of concept. The result of modeling or prototyping a business process using the ERP software functionality after it has been configured against your business requirements.

Prototyping. *See* conference room pilot.

Repetitive manufacturing. A form of manufacturing where various items are made across the same processes when production occurs. Products may be made in separate

batches or continuously. Production in a repetitive environment is not a function of speed or volume.

SCM. Supply chain management, as in a supply chain management system.

System. A collection of computer programs that allows you to perform specific business tasks. Some examples of applications are Accounts Payable, Inventory, and Order Processing. Synonymous with *application, or module*.

Tailoring. *See personalization.*

Acknowledgements

I wish to acknowledge and thank my former colleagues at the now defunct New Resources Corporation, who were my closest advisors and supporters of this project. Particularly Mr. Keith Danhoff for editorial and content review, Mr. Terry Horner and Mr. James Perez for their content review and alas, Mr. William Stagner and Mr. Norman Wold of the management team for their support of my efforts in undertaking this project.

All successful enterprise resource planning system implementations require great teamwork. No less effort is necessary when reducing the elements of project success into a written summation.

Author's Note: This work was originally published while I was in the employ of New Resources Corporation, a Chicago-based IT consulting firm, which ceased existence on December 29th, 2000.

About the Author/Presenter

Robert W. Starinsky is the Managing Principal of Tradewinds Group, Incorporated. In this role Starinsky serves as Senior Management Consultant and Industry Analyst for TradewindsGroup. Since 2001, Starinsky has been a Lecturer in the School of Computer Science, Information Systems and Telecommunications (CIT) at DePaul University in Chicago, Illinois where he facilitates graduate-level courses on business information systems, enterprise resource planning and project management.

Prior to his current position with TradewindsGroup, Starinsky had gained previous consulting experience, serving as a Practice Manager and Management Consultant with New Resources Corporation, a Chicago-based IT consulting firm that ceased operations at the end of 2000. Starinsky gained extensive industry experience prior to entering the consulting field in 1998. His diverse background includes manufacturing, distribution and financial services industry experience and he has served in a variety of managerial and professional capacities.

Starinsky has held important roles on numerous projects that have been directly related to the selection and implementation of packaged software solutions over the course of his 20+ year professional career. His software package experience includes knowledge of Enterprise Resource Planning (ERP), Customer Relationship Management (CRM), Computerized Plant Maintenance Management (CMMS), Supply Chain Management (SCM), Professional Services Automation (PSA) and various other software packages related to financial operations and management. In the ERP/CRM/SCM/CMMS space, he has worked with the products of many vendors, including those offered by ACCPAC, Best, J. D. Edwards, Great Plains, Oracle, P.S.D.I., Ross Systems, SAP and Siebel Systems.

Starinsky earned his Undergraduate degree at Northwestern University and holds Masters degrees from DePaul University (M.S.A.) and from Dominican University (M.B.A.). Starinsky also holds the professional designation of Certified Computing Professional (CCP). Starinsky is the author of two books, including Implementing J. D. Edwards OneWorld (Published in the Spring of 2001 by Prima Tech) and Maximizing Business Performance Through Software Packages (Published in the Fall of 2002 by CRC/St. Lucie/Auerbach).

Tradewinds Group, Incorporated is a management consulting firm serving organizations throughout the Chicagoland area, providing both general business and technology-focused management consulting services including expertise in and assistance with business process optimization and transformation, change management, software package selection and implementation, management auditing, project management and business/IT strategic alignment. TradewindsGroup serves smaller and middle market businesses engaging in manufacturing, distribution, construction and maintenance or professional service activities.

Revised for the Third Printing, Spring 2003.