GDI Consulting & Training
Making Breakthrough Changes at Breakneck Speed!

*Friends-of-the-Firm Briefing*

# Successful ERP Software Selection

*Proven Techniques for Improving the Information Systems Software Selection Process*

By
Alan G. Dunn, President
GDI Consulting & Training Company

**A Must-Read briefing for any team that is charged with selecting a new enterprise-wide software toolset**

*GDI Consulting & Training Company provides practical solutions to complex business and managerial problems. Focusing mainly on the manufacturing and distribution industries, GDI has developed a reputation as one of the most innovative and hardest working professional services firms in these industries.*

**GDI Consulting & Training Company**
P.O. Box 205 ● Temecula ● California ● USA ● 92593
Telephone 951-587-2003 ● www.gdiconsult.com ● agdunn@gdiconsult.com

# *Forward by Alan G. Dunn*

Dear Friends-of-the-Firm,

Although the debate over MRP systems versus Just-in-Time versus OPT and other forms of control has diminished within the last several years, there still exists a considerable interest in manufacturing software within the industry. The APICS MRP crusade of the late '70s and early '80s was highly effective in convincing many companies and practitioners that Material Requirements Planning (MRP) systems are a viable means of control and communication within a manufacturing company. In the past decade, many firms have installed fully functional ERP systems, and the debate continues as to how effective these systems are for the industry as a whole. It is not my intent to further engage in the debate of ERP vs. MRP vs. JIT (Just-In-Time) or other information management techniques, but to discuss issues inherent in the evaluation and selection of software systems, whether they be manufacturing, financial, program management or other.

**The purpose of this briefing is to explain and present proven techniques in the process of system evaluation and software selection.** There have been many articles written on software selection and generally accepted major steps taken to evaluate and install those systems. To some degree, most companies installing an information system follow those steps. It is not my intent to develop or present new methodology for software selection, but to discuss recommendations for software evaluation and opine on the effectiveness of those techniques. Along the way, I will question some common practices and offer some alternatives and recommendations. As the title of this briefing suggests, the step-by-step evaluation process that I will describe has been proven time and time again. I intend to discuss some of the problems I have seen companies encounter in trying to follow these steps and discuss particularly successful techniques that I have seen or used myself in evaluating software.

During the last 30 years, I have consulted for several dozen companies in all phases of their systems implementation efforts. These firms ranged from small, privately owned firms to some of the largest manufacturing corporations in the United States. The type of manufacturing environments ranged from pure process and repetitive manufacturing to job shop and huge project-oriented (aerospace and defense) type manufacturing. During this time, I participated in some remarkably successful implementations and some implementations that achieved few, if any, of the original objectives. **I have come to realize the degree of implementation difficulty often directly correlates to the effectiveness of the up-front systems evaluation and reasons for selecting specific software. The unfortunate fact is that many companies do a poor job of software evaluation.** The decisions are made based upon subjective data, personal attitudes, friendships and preferences. The company, software vendors, hardware vendors and consultants often intentionally foster this environment.

It is the intent of this briefing to outline how a company can make a logical evaluation of the current computer system and objectively evaluate new systems. The focus is on eliminating the emotional and intuitive aspects of this process and making a logical business decision. First, however, the company must make a commitment to conduct this type of evaluation and not allow the emotional attitudes to be established.

Enjoy your reading.

Alan G Dunn

Alan G. Dunn
President, GDI Consulting & Training Company

# Successful ERP Software Selection

## So You Want to Install a New ERP System?

Anyone who has been in the consulting business for any period is familiar with those companies that seem to make a never-ending project of evaluating new systems.  I'm not sure how this happens.  I suspect it is the result of personnel seeking to justify their existence or seeking to justify their budget.

**The first question the company must ask itself is why they are even interested in evaluating the current systems.**  Are continual problems being encountered by the Operations people and Production Planning and Control people in trying to obtain current information about job statuses?  Are schedules continually revised, yet shipments to customers are always late?  Are the inventories increasing but no one knows why or how much higher the inventory will climb?  In the accounting department, is information concerning material receipts and issues fragmented and incomplete?  Is the information required to calculate job profitability lacking and is the ability to properly cost manufacturing activity questionable?

In examining these problems, one must be careful to separate symptoms from true problems.  **The question must be asked whether these problems are being caused by an inadequate computer system or if the problems are caused by inadequately trained personnel?**  The lack of company policies, procedures and standard operating practices will cripple even the best computer information system.

There are generally THREE reasons why a company begins to seriously consider installing a new computer system.  They are:

1. The corporate or divisional headquarters decrees that a new computer system will be installed.

2. The current information system is deemed a failure, and the generally accepted solution is to replace it with another comparable computer system.

3. A company desires to upgrade to the next generation of hardware and/or software or convert from a manual to a computer system.  Typical examples are firms that wish to upgrade from an order point system to a Material Requirements Planning system, or firms wishing to upgrade from a basic MRP system to a fully integrated Enterprise Resource Planning (ERP) System.

The level of difficulty and therefore, chances of success of a new computer system directly relates to the reasons for change.  **Those companies installing a system because the corporate management has directed them to do so stand the biggest chance of failure.**  Local management and staff resent the intrusion from the corporate level and often feel their opinion and recommendations are not followed.  As a result, there is no "system champion" at the local level that will take responsibility for the new system implementation and make sure the project is run diligently and carefully.  In many cases, the corporate project team will supply internal personnel and outside consultants to assist and advise the local project team.  Usually these people are viewed as "carpetbaggers" at best or enemies at worst.  Despite the best of intentions at the corporate level, if this attitude is allowed to continue and foster, then the system implementation has the highest risk of failure.

**System implementations initiated to replace a system deemed a failure also pose an elevated risk of failure.**  Regardless of how poorly the current system is judged, there will undoubtedly be at least several people who were instrumental in installing the current system.  Those people will see the replacement of their system as an indication from the company that they have personally failed.  They may persist in proposing changes and enhancements to the current system long after management has decided it will be replaced and therefore delay or confuse the system project.

**Those companies installing a new information system with the intent of upgrading have the best chance of success.** Generally, most personnel recognize the inadequacies of the old system and are unanimous in their opinion that change is necessary. The implementation of the new system is viewed as a means of lessening their workload and providing more timely and accurate data. However, one of the common problems in this environment is that companies have usually devised elaborate manual systems to communicate the information or to conduct the check and balance validation. The challenge is to get people to step back and take an overview of the current practices. They must realize the waste and inefficiency inherent in largely manual systems and understand the greater efficiency promised by an automated system.

In considering the possibilities of implementing a new system, the company must always realize that a system is only a tool, not an end to itself. **No system can replace manual judgment or make decisions based upon invalid or missing information.** An information system is simply a business communication tool. If a current tool is inadequate because it cannot provide the necessary information, the promise of a new system is to provide new features and new reports that can satisfy the information needs. However, no computer system can compensate for bad data being input or mismanagement by uneducated users. In evaluating a new system, be wary of proponents who may promise that simply installing a new system will automatically solve a company's problems.

At this stage, management's role must be the creation of a project team and the appointment of a project manager. The focus must be on the objective evaluation of the current computer and manual systems and the formulation of recommendations. Personnel assigned to the project team must be capable of performing objective evaluations and be thoroughly knowledgeable of the operation of their respective departments.

**A key question currently is the extent of top management's involvement**. How many times have you heard the following quote?

> ***"Top management must be fully supportive and participate in the evaluation and implementation of a new computer system."***

Although no one can reasonably argue against this statement, it is rarely practiced. There are many reasons for this; top management is usually engulfed by the current business problems, they are often uninterested and confused by the technical jargon associated with computer systems, they are often alarmed over the high over-head costs of proposed new computer systems but resigned to the inevitability of that expenditure.

Senior management does not have to know the details of the existing or new computer system. **What is required of top management however, is an understanding of the advantages of a completely integrated formal manufacturing computer system**, understanding the need to translate a *business plan* into a *master schedule* and ultimately into *shop schedules*. I have yet to see a senior management team who, once they understood the advantages of an integrated information system, were not enthusiastic and supportive. What top management needs and deserves is an objective evaluation of the current information system and an objective "cost/benefit" evaluation of proposed changes or new systems. It is top management's responsibility to form and participate in a Steering Committee. It is the role of the Steering Committee to oversee the current system evaluation effort and to decide whether a new computer manufacturing system is to be selected and installed. It is also the role of the Steering Committee to review progress of the project team in the system evaluation, track expenditures against the project budget and (most importantly) fairly evaluate open business and system issues and make logical decisions based on the needs of the total company.

## Conducting the System Requirements Definition (SRD)

The next step is to conduct an honest appraisal of the current information system. The best means of providing that evaluation is through the System Requirements Definition. The objective of an SRD is to analyze each department's *current* and required information flow. The problems of the department in attempting to perform the job functions must be objectively analyzed and documented. The use and results of an SRD study are discussed in this chapter.

Once a company has concluded that the current systems are inadequate in meeting the needs of the firm, it must conduct a thorough evaluation of them. Many companies skip this step and proceed directly to the Request for Quote document or even by-pass the RFQ step and attempt to directly select a system based on perceived needs. It is these companies that become the perpetual system evaluators. With no specific data on what the firm's information needs are, and no objective evaluation of the current system's performance, they have no way of communicating to the software companies exactly what they need. System selection decisions are made based on personal friendships and intuition. Without any documentation on current system needs or definition of what the new system must be capable of providing, the chances of choosing and implementing the most appropriate system are remote.

Once the company has concluded that an evaluation is needed and the top-level Steering Committee has been formed to give direction and guidance, the company must form an evaluation team. This team should be comprised of knowledgeable "key" people from Manufacturing, Quality, Material, Design Engineering, Manufacturing Engineering, Order Entry, Marketing, Accounting Financial and Personnel.

The representatives on the evaluation team must be individuals who are thoroughly familiar with the workings of these departments and possess the capability of applying logic and common sense to the business system problems that will be examined.

After being formed, the evaluation team must start accumulating information on the current system capabilities and problems. There are three general means of gathering information in the Systems Requirements Definition process. They are:

1. **Develop a list of all system problems, concerns and needs from each department.** The list should include both computer and manual system related issues.

2. **Document the input, process and output for each department.** This entails the listing of every piece of information that comes into a department. That information may be contained in a computer report, memo or a telephone call. The means by which a department processes the information, the decisions which are made and the output (action taken or reports published) of the department is documented.

3. **Draw a flow chart for each department.** Using the structured analysis technique, the input, process and output of each department is documented on a flow chart.

All the above means of documenting the current system are valid and I have seen all of them used effectively. My own preference is to use a combination of number one and two, making a list of system problems and concerns and documenting the input, process and output. Once these are on paper, it becomes a straight-forward process of separating symptoms from the *true* root causes. Documented input, process and output lends itself very well to the formulation of a questionnaire to be used in the Request for Proposal. The documented information flow is compared to the information flow of a proposed software system. Exhibit I in the appendix is an example of several pages from a System Requirements Definition.

Although flowcharting has many advantages, especially in being able to pictorially present a complicated procedure, it does have its drawbacks. Often, a flow chart of even a simple process diagram can become exceptionally large and confusing. The time involved to construct and correct a flow chart can be excessive. Once the flow charts are completed and presented, department personnel may become confused and intimidated by the complexity of the charts.

When the System Requirements Definition is completed and the results documented, it must be presented to the Steering Committee and senior management. **The SRD must clearly and objectively present the problems and the advantages of the *current* system.** The next step is for the Steering Committee and senior management, working together with the evaluation team, to decide whether to replace the current system. At this stage, the company has one of two choices to make; either retain the current system and resolve the problems that have been identified or replace the system with a new one.

How can a company make a valid and cost-effective decision? The solution is to do a cost/benefit analysis. When considering whether to keep the old system, the company must evaluate the cost of enhancing and modifying the current system to solve the problems which have been identified and the estimated time frame of the modification. Besides solving the system problems, nonexistent or outdated documentation must be updated. When considering the cost of enhancing the current system, the cost of re-doing the documentation is sometimes ignored, yet can exceed the cost of enhancing the software system.

Costs and benefits for installing a new system must also be calculated. Costs include software, possibly hardware, training of the entire company on the new system and implementation costs for special programming and consulting. It must also be recognized that a typical implementation time span is from one to two years. It is a sad fact that many companies leap into the decision to buy a new system without completely calculating costs or considering the time involved in installing it.

If the decision has been made to keep and enhance the old system, the following activities must be organized:

1. A re-implementation team needs to be organized, preferably using the same people who took part in the evaluation. It is these people who have done the in-depth evaluation and are in the best position of recommending solutions and working on the implementation.

2. As with any project, a detailed project plan must be developed with major milestone dates and a critical path identified. Major milestones of a re-implementation are like those of a new system implementation.

3. System analysis, programming and testing effort must be calculated, both in terms of cost, and availability of manpower. In addition, the time and cost associated with retraining existing personnel must be determined.

Several years ago, I consulted with a company going through a re-implementation of their software package. The project team member in charge of education had designed a sign that hung on his wall that was a derivative of the famous Ollie White saying, ***"If you think education is expensive, try ignorance."*** This individual had a sign that read, ***"If you think education is expensive, try re-education."*** That saying underscores the fact that breaking poor system utilization habits is a time-consuming task.

Once the programming tasks and re-education are completed, and the implementation of the enhanced system is ready, the implementation cutover must be planned as carefully as though for a new system. Conversion programs may need to be written and scheduled and new database information may need to be loaded. The last step of a re-implementation is to define performance measurement reports. The purpose of the performance measurement report is to provide department management and senior management with concise information on how well the new system or enhanced system is being used. This type of performance information is crucial for a company to avoid the system problems that may have happened in the past.

If top management decides to replace the old system with a new system, a subsequent decision must be made to either write a new system or to buy a "canned package". This can often be an emotional decision. If the current system was developed internally, the Data Processing department may see management's decision to buy canned software as an insult to their department. Many companies, including some major manufacturing firms, still have the attitude that anything "not invented or developed here" is inadequate. Data Processing personnel may view the decision to buy a software package as a threat to their job security and occasionally are justified in thinking that. When debating the question of buying a software package or writing the new system in house, the decision must be made using objective information. The major advantages and disadvantages of buying canned vs developing in house is discussed below.

The major advantage of developing a software package in-house is that the company personnel know best what specific features are needed. The package can be custom tailored to support whatever unique requirements are needed by the company. In this type of environment, a nearly 100% fit should be an obtainable goal.

The major disadvantage in developing a system is the number of highly skilled (and expensive) technical people necessary to construct a large system. The cost of this can easily become an excessive burden for any company. System Analysts, System Programmers and Application Programmers are needed to design and construct the system. The time involved in doing this may stretch into years and easily cost millions of dollars. Once this system is designed and written, adequate documentation must also be written.

In buying packaged software, the major advantage is the ability to begin immediate implementation of a large comprehensive package. The software development time and testing are a minor concern. The software vendor has already designed, debugged and documented the system. The major effort is to educate and train the users and implement the software system.

The major disadvantage of canned software is the fact that the software system has been designed to be applicable to as many manufacturing companies as possible. Therefore, a general rule of thumb is that an 80% fit is the best that a company can expect. The additional 20% must be supported either by programming enhancements or through the use of report writers and other features.

Once the company has concluded that the best and most logical choice lies in purchasing a new software system, a thorough evaluation of all possible software systems must be conducted. This is no small job, considering the considerable number of systems currently on the marketplace. The next chapter discusses the design and writing of a Request for Proposal questionnaire to be used in evaluating software packages.

## Request for Proposal

Using the information obtained in conducting the System Requirements Definition, the company must construct a questionnaire to be included in a Request for Proposal (RFP) to go out to the software vendors. The SRD documented the basic information flow which must be supported by the computer system. The RFP questionnaire identifies specific features and functions that the company is looking for in the new system.

Through answers on the questionnaire, each software package can be evaluated as to how well it supports both present and future needs. In constructing the questionnaire, the company should not be concerned whether the desired feature is available in any package. It is perfectly acceptable to do a little "blue sky dreaming" when constructing the RFP. The intent is not to have a software vendor achieve a 100% score, but to determine which software vendor comes the closest to satisfying most of the needs.

The company should consider obtaining the services of an outside consultant to help it construct the questionnaire. If internal personnel are not familiar with the software industry and software systems that are available, much time can be lost in simply determining what is currently available in the marketplace. **A consultant who commonly performs software evaluations can provide valuable help in explaining the type of systems that are available, the type of features that are common and the major software vendors.** However, in selecting a consultant, a word of warning is in order. Whereas most consultants have earned a reputation of being honest and objective, the fact remains that there are some biased consultants. There have been instances in which consultants steered clients toward a software package that was not appropriate for them, due to the consultant's own personal preference or hidden business relationships with software vendors. Software vendors try to maintain good relationships with consultants to the extent that the consultant is familiar with their product. But this relationship should not go so far that the consultant is more interested in the software vendor than in the needs of the client.

When constructing the RFP questionnaire, the company must require that the software vendor answer with more than a simple **yes** or **no**. The software vendor should be instructed to enter one of a minimum of four answers. The possible FOUR answers are:

1. The standard software package fully supports the desired function.

2. The desired function can be supported through a report writer.

3. The desired function will be written in the future as a planned enhancement.

4. The desired function is not available in the system.

I have seen many Requests for Proposals in which a fifth answer was, *"Current system can be modified to provide feature. Include estimate of modification costs."* I completely disagree with this type of response. In my opinion, the software vendor cannot possibly provide any firm cost estimates of modifications from the RFP. In addition, providing customized software modifications to clients creates a number of serious side effects such as documentation for the modification and continuing software support from the vendor. It is *only* appropriate to discuss custom modification to the software *after* the company has entered serious negotiations with a software vendor. Exhibit II in the appendix contains several pages from a sample Request for Proposal.

In addition, the software vendor should be encouraged to include additional supplemental information to further explain the system that is being proposed. Many software vendors can offer creative solutions using their system to solve a number of system needs. The software vendor should be able to fully and completely describe the proposed system on the RFP questionnaire.

In addition to the questionnaire, the Request for Proposal should require the software vendor to provide background information about the company and the products and services it offers. The software vendor should be asked to provide information on the length of time the company has been in business and give an indication of the number and types of systems that have been sold to date. The vendor should include financial information to indicate the stability of the company. Development plans that are scheduled for the future should be described and include an overview of their overall software development goals. The software vendor's support and educational offerings should be completely described as to class locations, cost and schedule.

Many firms engaged in the search for software spend considerable time talking with representatives from software companies, listening to their proposals and sales presentations. This type of software evaluation creates many dangers. Primarily, it does not adequately ensure that all possible software candidates are fairly evaluated on the same basis or are given an equal opportunity to make presentations. In this type of environment, software salesmen primarily rely upon developing inside salespersons. This is the technique of grooming and supporting the internal person or persons who are the leading supporters of that particular software system. As a result, the internal software evaluation team becomes divided into factions, split by loyalties and personal friendships that have been fostered by the software salesman.

In order to provide equal time to all software vendors and also avoid the conflict of personal contacts, I strongly recommend the scheduling of a **bidder's conference**. The RFP is distributed at the bidder's conference and discussed rather than simply mailed to each software vendor. In a bidder's conference, all potential software vendors are invited to attend. This minimizes the initial contact with software vendors. Any vendors that contact the company or any candidates that are contacted by the company are requested to forward sales information and the business card of the person who should receive the invitation to the bidder's conference. All other contact should be discouraged in the interest of not allowing personal relationships to interfere with an objective evaluation. The only exception of avoiding personal contact is in the situation of meeting with a vendor who is not sure if their system is a possible candidate. This meeting should be strictly limited to providing information about the company and its environment, as it relates to the search for a new system.

The bidder's conference should have the following agenda:

1. A high-ranking official of the company, preferably the Vice President of Operations or the General Manager, kicks off the meeting with opening remarks. Those remarks consist of expressing appreciation for those who have attended, some high-level explanation of the project, an introduction to the company and its products and an introduction to other company employees attending the meeting. At that time, sales literature or brochures that explain the company's products should be distributed.

2. The next person to speak is the Project Manager. This person will give a more detailed explanation of the system implementation plan, explain the project team organization's roles and responsibilities and provide the background of events that led the company to decide to select a new software system.

3. The Request for Proposal documents are distributed among the attendees. The project team members, with support from the consultant involved (if requested), will then review the Request for Proposal. The major highlights of the RFP should be discussed with any level of needed explanation provided. Key points to emphasize are contact person who should receive telephone calls concerning the RFP, due date for responses, type of literature and sales information which the software vendor should provide and the grading scheme to be used in answering the questions. Each section of the RFP should be reviewed, highlighting and discussing the major points. Questions from the attendees should be encouraged as the subsequent discussion will be beneficial to all. A primary focus in discussing each section should be to explain the current situation within the company and the reasons why a particular feature is required. This type of information is valuable to the software vendors who will be attempting to respond to the questionnaire. While the questions provide the primary input as to what is needed by the company, communicating as much background information as possible helps the software vendor fully understand the scope and focus of the project.

The software vendors should be allowed a reasonable period of approximately 3 weeks to respond to the RFP. Less than 3 weeks does not allow adequate time to provide a quality answer to the RFP questions, particularly if it is a lengthy RFP.

When the RFP responses are received from the vendors, the vendor overview information, i.e., vendor qualifications, length of time in the manufacturing software business, qualified vendor list, financial reports, etc., should be removed and reviewed first. This type of subjective data is hard to categorize by a simple numeric-rating scheme. However, this type of vendor-supplied information, particularly financial stability information, should be reviewed and vendors disqualified immediately if necessary. In grading the responses from the RFP, I recommend the use of a spreadsheet type program. Each question number can be loaded in the spreadsheet and a weighting factor assigned to the answer. The weighting factor, which is determined by the responsible selection committee member, determines the importance of the particular feature that has been requested. The numeric answer of each specific vendor is then also loaded into the spreadsheet. The score for each vendor is calculated by multiplying the weighting factor times the vendor answer. The scores are then totaled and compared to the highest possible score. By using the spreadsheet, it is possible to calculate standard deviations and variances between the leading software vendors. Any vendor that scores unusually high should be identified for additional investigation and evaluation. Exhibit III in the appendix contains several pages from a spreadsheet recording responses to an RFP.

As determined by this grading scheme, the leading two to three vendors should be selected. Vendors who are not qualified should be sent a polite letter explaining that they have been disqualified and an explanation as to the reason why. If a vendor has taken the time to attend the bidder's conference and respond to the RFP, they deserve feedback on why their system did not measure up to the other systems evaluated. The leading two to three software vendor candidates must be contacted and informed that they have been selected as finalists in the evaluation. As with the case of those disqualified, these vendors should be told in what areas their system or company was judged to be superior. At this time, these vendors must be invited to the plant for a face-to-face meeting with the project team members. The next step of the process, scheduling extensive software demonstrations for the project team, should be completely explained and discussed to ensure a thorough understanding. The type of software evaluation that is described in the next chapter can easily take 2 to 3 full days. The process consists of providing basic database information to the vendor to be loaded into the vendor's demonstration system and a complete beginning-to-end demonstration is conducted. A specific date will be arranged with the vendor and the appropriate company evaluation team members scheduled to attend. At this time, the evaluation team's responsibility is to develop specific questions and scenarios to be presented to the software vendor during the evaluation. The process of developing the detailed scenarios and participating in a software demonstration and evaluation is explained in the next chapter.

## Vendor & Software Evaluation

In evaluating a software vendor's strength and capabilities, four areas must be examined. They are **technical competence, presence in the marketplace, financial stability and size and strength of support staff.**

## Technical Competence

Unfortunately, there are no definitive means of evaluating a software vendor's technical capability. However, there are several indications such as the percentage of revenue devoted to research and development or the size of the development staff in proportion to the total employment of the company. However, those types of numbers are usually confidential and nearly impossible for an outsider to determine. It is easy for a vendor to overstate or mislead the questioner when this type of information is sought. Other means of determining a vendor's technical competence can be to; measure the average number of problem notices in the past year, measure the number of average calls per day to the customer support hot line and, more effectively, ask current clients of the software firm for their evaluation of the software firm's technical ability and responsiveness to the problems reported. In recent years, determining technical competence has become further complicated because of the increasing number of mergers and buy-outs which are occurring in the software industry. Often, the software system being proposed by the software vendor was developed by another firm that was acquired by the company. In this situation, the questions that should be asked concern the number of original development people still on staff.

## Presence in the Marketplace

The software firm's presence in the marketplace is a good indication of the acceptance of its product. Most, if not all, software firms will be eager to share this type of information. In addition, industry journals and magazines often conduct market surveys and trends and publish the results. Your data processing staff should have some of these periodicals available. One leading magazine conducts yearly surveys of its readers on the type of software system they are most likely to purchase.

## Financial Stability

The financial stability of the software firm is an important aspect which should be evaluated but may be difficult to determine since many software companies are privately owned and therefore do not publish their financial information. When contacting each software firm and asking for a response to the RFP, each firm should also supply you with a copy of their 10K form. The 10K includes the firm's income statement and balance sheet for the previous fiscal year. Your firm's accounting management should review the 10K forms and request the Dunn and Bradstreet credit rating. However, an unfavorable financial report for the previous year should not automatically disqualify a software vendor. The software industry suffers from many peaks and valleys in revenue. Only if a software firm appears in imminent danger of declaring bankruptcy should the software firm be disqualified. Otherwise the financial performance over the most current 3 to 5 years period should be used as the basis for determining financial strength.

## Strength of Support Staff

The size and scope of a software vendor's support staff should be evaluated. The size and frequency of educational classes should be obtained. The location of the nearest support office should be obtained and a rate schedule of fees that are charged should be included in the software vendor’s response to the RFP.

Software vendors will often invite representatives of a perspective client to visit the company's corporate office for a day or two to discuss the vendor's qualifications. These types of meetings must be seriously considered and attended if possible. Only by meeting face to face with the software vendor's senior management can a prospective client feel comfortable that the information received from the vendor appears to be valid.

## Software Evaluation

After the vendor has been evaluated, the next and very crucial step is to perform an extensive evaluation of the proposed software system. There are several means of conducting an evaluation of the software system. Software demonstrations can range from a brief 2-3 hour visit to the vendor's office to a complete trial installation at a prospective buyer's site with months of time and effort spent by the project team to learn the system and input transactions. Except for the very largest companies, the latter means of evaluation is prohibitively expensive in terms of personnel time and expense. The brief 2-to-3-hour office demonstration cannot possibly cover the major features of the system.

I recommend that the software vendor conduct an extensive 2-to-3-day demonstration at the vendor's office for the software evaluation team. In order to accomplish a profitable evaluation, much work must be done by the evaluation team in advance of the scheduled demo. The first step is to request and secure a commitment from the software vendor to conduct an extensive demonstration using the prospective buyer's database information, i.e., part numbers, Bills of Materials, shop routings, customer orders, etc. The evaluation team then writes a comprehensive overview of the types of transactions that they expect to see during a software evaluation. The overview scenario should encompass a complete "womb to tomb" flow of information through the vendor's software system, beginning with an empty database, and ending in the shipment of finished goods to a simulated customer. The overview scenario is provided to the software vendor for their review and comments.

The next step is to select the **real** product structure to be used for the demonstration. The evaluation team must be careful to select a representative product, but one that does not have too many parts or have too many BOM levels. The data must be kept to a minimum to prevent excessive time spent by the software vendor simply to load data. A product structure of 10-15 parts in a Bill of Material no more than 3 or 4 levels deep should be used. This product structure can be a sub-assembly of the final product, or it can be an abbreviated version of a production Bill of Material. This database information will be given to the vendor to load at the beginning of the demonstration. Since most manufacturing software systems today feature full, on-line real time input of transactions, this should not pose a problem. However, if the software system being proposed includes many batch transactions which require all data entry to cease while batch programs are processed, then this approach may not be feasible. It might be necessary to provide the software vendor with the database information in advance of the demonstration to allow the vendor to load it prior to the evaluation.

Once the evaluation team has selected the product structure to be used for the demonstrations, they must then familiarize themselves with the software system to be reviewed. Nearly all software firms offer a virtual mountain of sales literature and information for their prospects. Most of this information contains specific information on the software system.

The next step is for the evaluation team to write a detailed demo script to be used in the actual demonstration using the overview as a guide. The specific questions should be designed to evaluate the vendor answers on the Request for Proposal document. I find it ludicrous to see a company spend days or weeks preparing a Request for Proposal, yet, in conducting the actual software demonstrations, virtually ignore the requirements that were given to the vendor. The demo script should be a logical sequence of events that the vendor must follow in demonstrating the specific functions of his system. Using the overview script as a guide, the evaluation team member responsible for each specific area develops a detailed script for each functional department. For example, the Purchasing Department software evaluation team member should prepare a detailed demo script which will cover the following transactions: loading the vendor master and detailed descriptive information, inputting a purchase order, processing a return to vendor transaction and determining the specific type of hard copy purchase document which the system provides. In short, the evaluation team member is responsible for accumulating all the information possible in terms of system functionality in his area of responsibility. Next, the actual demonstration is conducted. Since the overview demo script was provided to the vendor in advance, the software vendor should be prepared to provide the type of demonstration needed. As the demonstration begins, each evaluation team member then uses their own demo script to ask specific questions about the system functionality and provide guidance to the software vendor personnel about the type of transactions that need to be demonstrated. The evaluation team members must keep careful notes on the information that is obtained. Whenever possible, sample reports should be collected from the software vendor, be kept as reference and also compared with the answers as understood by the evaluation team member. At the conclusion of the demo, each evaluation team member should have a comprehensive understanding of the software system functionality. A debriefing meeting is conducted shortly afterwards to allow each team member to compare notes and ensure that everyone has the same understanding of the system. The responses to the demo script can be compared to the Request for Proposal. In this way, misleading answers that were indicated on the RFP can be discovered.

One last means of evaluation is to visit companies that have installed the software firm's system. However, it has been my experience that user visits are generally a waste of time and money. First, the software vendor will only reveal the names of companies that are their best users, not the average or poorer use. I recommend that several current users be contacted by telephone. The focus should be on the number of software "bugs" encountered, the responsiveness of the software firm in solving the problem, and the expertise of the software firm's support staff.

Finally, (as stated previously) don't expect more than an 80% fit of any software package. Remember, the vendors have developed the package to be as broadly applicable as possible to a wide range of companies in different industries. Though most software firms do a decent job of building options and flexibility into their system, it isn't possible to develop a packaged system that can completely satisfy the needs of "a" company. The job of the evaluation team is to select the software system which comes the closest to meeting the company's needs. They must understand that the last 20% is achieved through report writers and special programming which must be done by each user company. To criticize the software vendor for not offering a complete 100% solution is unfair and unrealistic.

While the above evaluation process is rather time consuming and lengthy, especially for the evaluation team which must develop demo scripts and become familiar with the system, I am convinced it is the most economical and productive means of evaluation. It is certainly more comprehensive than the 2–3-hour types of demos that are common. It is also less expensive and more efficient than installing the system on your own computer for a trial installation.

Another issue that may arise during an evaluation process is the question of system integration versus interfaced systems. This question arises from the fact that many manufacturing systems do not contain completely integrated manufacturing, accounting and (if required) program management systems. Most major software vendors have made great strides in developing truly integrated manufacturing systems, yet a potential buyer of these systems may find an integrated system judged to be superior in one area but extremely weak in another. On the other hand, a software vendor proposing a system offering only manufacturing capabilities may be judged as having the strongest system in that area but cannot offer a financial package, much less an integrated system. What is the best solution to this type of problem? A firm in this predicament has one of two choices: buy the integrated system and make do as best as possible with its weak parts or buy the best software package in each area and develop the interface programs in-house. The choice is not easy, and it often depends on the company's available resources. In the final analysis, the company must judge the fit of the best fully integrated package versus the technical expertise available in-house to write software bridges between the different systems.

Once the vendor and the software system have been fully evaluated, the evaluation team must select the leading candidate. The steps in achieving this are described in the next chapter.

## Final Selection

By this time, the evaluation team has put considerable time and effort into the following FOUR evaluation criteria:

1. The software vendor has been evaluated in terms of presence within the marketplace, popularity of the software product proposed and financial stability and post-sales support.

2. The proposed software has been ranked against the defined company requirements using the System Requirements Definition and the Request for Proposal as a guide.

3. The evaluation team has participated in extensive demonstrations and has made notes concerning their approval or disapproval of the software systems.

4. The evaluation team has spoken with other companies using the vendor's proposed software.

From the above criteria, the evaluation team should be getting a clear understanding of the true strengths and weaknesses of the various proposed finalist systems. The evaluation team then conducts a round table discussion of each specific software vendor and system. Each team member is responsible for sharing their impressions concerning the vendor and software. All strengths and weaknesses of the proposed system must be aired and discussed. To aid in this evaluation, the evaluation team may elect to develop a score sheet for each vendor. Each evaluation team member's concerns and recommendations should be listed on the score sheets.

By this time, unless a severe division has occurred within the evaluation team, a gradual consensus should be forming among the team. However, to officially select the leading candidate, a formal vote should be taken among all members of the evaluation team. The following parameters should be used in conducting the formal vote:

- Each evaluation team member has only one vote.
- The goal of the evaluation team should be to reach a consensus, not to attempt a simple majority.
- Each team member must consider the needs of the company, not just the needs of each team member's specific department.

After considering all the considerations, the evaluation team makes a final selection of the best system. In actuality, the evaluation team members should select the top two leading candidates. If the company is unable to reach an agreement with the leading software vendor, the next step is to initiate negotiations with the runner up.

In negotiating with the software vendor, your company's most experienced purchasing and legal personnel must be called upon. Software agreements are often complex and confusing, with agreements among second-tier firms often involved. In negotiating the software vendor's price, your company must remember that the task of determining the cost of the software system is often very subjective. The actual cost of the software is minimal, consisting only of magnetic tape and labor involved to make a copy. However, behind that software system is many man years of programming and testing. The software vendor has priced his software to recoup part of the investment that was spent in developing it.

In negotiating with the software vendor, be aware that you are really negotiating the price to license the software, not to purchase it. What is being sold is the right to use the software at your plant for a specified amount of time. Software agreements usually are specific about the type and the location of the computers to be used to process the system.

In addition to the initial licensing fee, you will also be expected to pay the annual maintenance fee. That fee includes sustaining support for additional software enhancements or modifications, and the use of a customer service hot line at no additional fee.

The vendor usually determines the price of the software support depending on the size of the computer and the number of peripheral devices the company expects to be running. If you increase or upgrade to a larger computer or add additional devices, most software agreements require you to pay a higher annual license fee, even though the software system is exactly the same. When negotiating with the vendors, get an explicit definition of what the pricing levels are and, when possible, limit any automatic fee increase due to the size of the computer.

Many software companies include education and services with the licensing fee. When negotiating, insist that all fees be listed separately and insist on the right to select among the services you feel your company will require. While many of these services and educational offerings are worthwhile, you must retain the right to use only those that are appropriate.

The final step is the formal agreement and signing of the contracts between your company and the vendor. The goal is to establish a good business relationship that will continue for many years. Be careful not to conduct your evaluation with such an antagonistic tone that relationships between you and the vendor are poisoned. During the initial implementation, the expertise and system knowledge of the software vendor will be crucial. The last chapter of this article will further define the types of services you can and should expect from the software vendor.

## Continuing Support

During the past 25 years, there have been numerous books and articles written on successful ERP implementation. The topics discussed, such as top management involvement, the formation of formal implementation teams, and ongoing education and training plans, are all worthy activities I fully support. It is not my intent to describe a complete ERP implementation in this article. What is my intent, however, is to describe how the initial work of documenting the system needs and developing a Request for Proposal can continue to be used during the software implementation.

One of the things that has always disturbed me in collaborating with companies installing ERP systems was that after the software is selected, a transformation seemed to take place. A new team of people was selected that often began from nothing with the implementation. All the previous work, such as the System Requirements Definition and the Request for Proposal, was put on the shelf and forgotten. This is incredible! Those documents were prepared as the company's definition of what was needed in terms of the system required to run the business. It is a tragedy for the implementation team to ignore the demanding work and the results of that effort. By not understanding the company's core information needs and system problems, the implementation team often becomes lost in a maze of the detailed implementation tasks. Often, they believe their only obligation is to install the system, ignoring the original goals and objectives that were to be addressed. An implementation team that has lost sight of the core objectives will soon become hopelessly lost in petty issues and squabbles. It is these types of implementations that result in the horror stories of companies working for 6 years on a system implementation project with little accomplishment.

The initial System Requirements Definition must be used to provide the guiding implementation direction for the implementation team. Communication gaps and information needs as identified in the SRD become the central objectives in installing the software. A check list should be defined for each department which lists information needs and how the new system will fulfill those needs. A means of measuring progress towards reaching each goal must be developed.

As policies and procedures are defined, they are done in accordance with the System Requirements Definition. Once all new policies and procedures have been agreed upon, the final step is to update the SRD documents to show what the new information flow is using the new reports and the new system. The results along with the ERP system should be a current System Requirements Definition listing all information flows and features of the new system that are used.

## Summation

The final advice I have for companies installing ERP systems is to stay in contact with the software vendor. As questions arise concerning the software system, not only during original implementation but later, the software company is the leading source of information on how other companies have used that feature. Also, the software vendor is the source of information as to what enhancements for the system are planned in the future and when they will become available. Finally, in today's competitive marketplace, software vendors are looking for show case clients to host other prospects. It is in the vendor's best interest to see that your company is successful with the new software. They are hoping to eventually be able to use your company as a reference and to show off their system to its best advantage. Rather than discourage this type of visit, you should take advantage of it. This is your opportunity to demonstrate to other companies what you have accomplished and pass along your advice and recommendations on how to successfully install a new computer system.

# About the Author

Alan G. Dunn is currently President of GDI Consulting & Training Company and founder of the Manufacturing Executive Institute (MEI). He is also the creator and lead-instructor of the 18-month Next Generation Global Supply Chain Leadership Development Program at the California Institute of Technology's (Caltech) Center for Technology & Management Education (CTME), where he has taught since 1984. Mr. Dunn also serves on the University of California at Riverside's (UCR) Advisory Board for Transformative Leadership in Disruptive Times.

Mr. Dunn specializes in supply chain management, strategic planning, manufacturing management, operations management, leadership development, cost management and business finance.

Previously, Mr. Dunn was a Vice President at Gemini Management Consulting and a Partner at Coopers & Lybrand. In both positions, he led large technical manufacturing teams through innovative productivity enhancement projects. Mr. Dunn has participated in >188 significant manufacturing and distribution projects inside >118 companies. He has worked in 24 countries and across most manufacturing sectors.

Over his 40-year career in global supply chain consulting, Mr. Dunn has served on the Boards of Directors of numerous public, private and non-profit companies. He is the recipient of the National Association of Corporate Directors (NACD) prestigious *"Director of the Year"* award in 2007.

Alan is a career Association of Supply Chain Management (ASCM) volunteer, having served as the President of the Orange County Chapter in 1984 and Chairman of ASCM in 2015. He was inducted into the *"ASCM New England Supply Chain Conference Hall of Fame"* in 2022.

Mr. Dunn has a degree in business management from California State University, Fullerton.

# Contact Information

**Alan Dunn, President**<br>
**GDI Consulting & Training Company**<br>
P.O. Box 205<br>
Temecula, California 92593 USA<br>
951-587-2003<br>
agdunn@gdiconsult.com<br>
www.gdiconsult.com