

---

# Clinical Trials Management System (CTMS)

## Guide for Evaluation and Selection





Protocol Manager™  
©Winchester Business Systems, Inc.  
304 Cambridge Road  
Woburn, MA 01801, USA  
[www.wbsnet.com](http://www.wbsnet.com)  
All rights reserved

©Winchester Business Systems, Inc. 2005. An unpublished work subject to trade secret protection. This work contains herein confidential and proprietary information of Winchester Business System, Inc., and should not be copied, circulated or distributed to persons not employed by Winchester Business Systems, Inc. Unauthorized disclosure of this work is expressly prohibited. Information presented and contained herein is subject to change without notice.

# Preface

---

Recent technological advances have led to the development of many new hardware/software and manual devices, each relevant to the planning and control systems employed by Life Sciences companies. Management, virtually inundated with options, is confronted with the problem of determining which systems to buy. Although hardware/software sales literature and vendor's web sites reveal differences in competing products, the significance of these differences can be unclear.

Management, and even the consultants it employs, still needs to determine what specific components are needed, how many, how they should be deployed, and how they should be integrated with existing systems to form a complete Clinical Trials Management System (CTMS).

Clinical management is supplied in this paper with information useful in evaluating the company's CTMS needs, and in selecting a software vendor and package that best fits those needs.

For those clinical organizations that, without outside assistance, decide to install a CTMS, this paper provides help toward achieving benefits more quickly and at substantial savings.

This paper is addressed to senior management and the project team that is charged with implementing new systems. It describes a rational method for the evaluation and selection process, and indicates ways in which implementation projects can be structured to yield benefits and supply appropriate controls to ensure those benefits.

There are many ways to accomplish the evaluation and selection tasks. We here provide an approach that will prove to be useful to those organizations desiring to undertake such a project. The information is broad in nature, designed to serve various Life Sciences industry environments.

The approach discussed here is one of several approaches to evaluating and selecting CTMS software.

Winchester Business Systems, Inc.  
2006

# Contents

---

Preface .....	iii
Software Packages are Viable Alternatives .....	1
Basic Steps .....	1
Seven Steps to Selection .....	2
Selection Analysis .....	4
Step 1 – Establishing Objectives and Criteria .....	4
Step 2 – Classifying Objectives According to Importance .....	5
Step 3 – Developing Alternatives .....	8
Step 4 – Evaluating Alternatives Against the Objectives in Order to Make a Choice .....	8
Step 5 – Choosing the Best Alternative as the Tentative Selection .....	11
Step 6 – Assessing the Adverse Consequences of the Tentative Selection .....	11
Step 7 – Controlling Effects of the Implementation of the Final Selection by Preventing Adverse Consequences and by Follow-up .....	13
Keys to Implementation Success .....	14
Winchester Business Systems .....	15
Bibliography .....	16

## Software Packages are Viable Alternatives

---

Vendor-supplied software can provide a cost- and time- beneficial alternative to developing 'tailored' in-house systems. Over the past ten years, independent software houses have sprung up nearly geometrically, offering a myriad of options to potential buyers. In addition to independent software houses, the vendors of database systems and electronic data capture (EDC) are now offering CTMS software alternatives directly or through business partners.

By now there must be, among the hundred or so options for CTMS software, at least one that best fits a particular organization's system requirements -- but which one? The cost of making the wrong selection can be enormous.

There are numerous case studies that describe mistakes made by organizations, both in developing in-house systems and in implementing purchased software.

A principal mistake has been for a company to buy a CTMS to manage the current, low-level of clinical trial activity. When the company expands from Phase II to Phase III trials, the volume of documents and data to manage seems to grow exponentially. These companies have often found that the simpler, often ASP-hosted systems they originally deployed, will just not handle the new level of volume.

It was this inefficiency that attracted the attention of Winchester while working with clients. For several years, Winchester has been advising clients on their system requirements and what pitfalls to avoid.

Winchester consultants observed that a great many software selections were often poorly made and ultimately extremely costly.

The consultants were convinced they could provide principles and basic techniques to improve this management decision-making process.

### Basic Steps

Using some basic steps and methodology derived from problem-solving theory, Winchester's consultants developed the following process to evaluate and select computer software. The validity of the process has already been established in the most convincing way by practical application *on the job* in numerous client engagements. In order to make the right software selection, the evaluators have to know how to ask the right questions. It is precisely with the selection process that this paper is ultimately concerned.

## Seven Steps to Selection

---

The basic steps to evaluating and selecting the appropriate software alternative follow a rational progression.

The design of a system produces a solution to a management problem that can be verified because the problem already exists. However, the selection of best fit alternatives provides a solution that cannot be verified because its implementation will take place in the future. Nevertheless, the project team making the selection has to choose an alternative of one kind or another. If the team members have systematically identified system requirements, they will be in a position to make sure that the alternative they select will actually satisfy those requirements, temporarily or permanently. Making the best selection involves a sequence of procedures based on the following seven steps:

1. The objectives, systems requirements, or criteria must be first established. What is the implementation of the system trying to accomplish? What is the job to be done?
2. The criteria are classified as to importance. The selected software that fits the computer hardware installed would be a **Must** requirement that cannot be compromised. Accomplishing these objectives at low cost and implementing quickly are categorized as two **Wants**.

Wants are not requirements; they are open to compromise where the client would like to have the best performance possible. The Wants are then ranked and weighted; e.g., if low cost were more important to the client, this Want would be more important in the final decision than would the timeliness of implementation.

3. Alternative software packages are determined. With the proliferation of hardware and software systems, there are usually many ways of satisfying the listed criteria and getting the job done. Some options are cheaper or better than others. Software reference libraries are full of advertising materials and catalogs that describe vendor offerings. It is likely that a visit to one or two of these libraries, a review of trade publications, and perhaps published software evaluations will provide a significant number of alternatives and vendor offerings.
4. The software alternatives are evaluated against the established criteria. Each alternative is assessed as to whether it satisfies each of the Musts (e.g., the software will run on the client's hardware) and as to how well, relative to each other alternative, it achieves each of the Wants (e.g., low cost, timely implementation).

5. The choice of the software alternative best able to achieve all the criteria represents the tentative selection. The best software alternative meets all of the Must requirements and matches most of what is wanted with the fewest disadvantages. It is the software that, on balance, satisfies the requirements the best. The choice or selection may call for a combination of alternatives. For example, if the client requires several subsystems, possibly the best solution would be to select one application from one vendor and the others from other vendors.
6. The tentative selection is explored for possible future consequences. A problem could result from a planned upgrade or switch in computer hardware; i.e. 'scalability.' Such a problem is assessed as to probability and seriousness. If the difficulties resulting from a contemplated selection seem to be too great, another decision or selection may be necessary. This step also assists the project team in choosing between two alternatives that were very close during the selection process in Step 5.
7. The effects of the final selection are controlled by taking actions to prevent possible adverse consequences from becoming problems, developing an implementation plan that describes the steps to follow, and by making sure the designed steps are carried out through proper project management. Thus, the adverse consequences are minimized or avoided by:
  - ☞ anticipating problems in implementation before they occur;
  - ☞ developing a plan of action that describes the best steps necessary to effect timely implementation;
  - ☞ assessing the resource requirements to carry out the plan; and
  - ☞ implementing the system according to the plan.

This summary of the seven basic steps in software evaluation and selection is intended to show the form and application of each step. The process described above involves stating clear objectives and carefully evaluating alternatives -- all aimed at taking action.

The method outlined in this paper cannot assure the project team of a successful software selection. No method can do that. But this procedure should help members of the team to assess the facts available and to capitalize on their own experience in making the final decision. The more systematically they can go about this, the more chance they have to bring their experience and good judgment to bear.

## Selection Analysis

---

Now, let us look more closely at each of the seven steps in this software selection process and see how they relate to each other and contribute to the decision making process.

### ***Step 1 – Establishing Objectives and Criteria***

This extremely important step is often omitted or superficially handled in software selection. Often, only the broad idea of the objective, such as ‘control inventory,’ is stated. Such an objective is too vague to be of much use as a standard of comparison in choosing the right inventory planning and control system. The objective or criteria must be specific. What inventory should be controlled? What planning rules should be followed? How does inventory flow? What transactions that affect on-hand balances should be tracked? What SOPs are affected?

The objectives should describe the goal precisely and locate it in time, place, and number. Thus, ‘control inventory’ becomes ‘account daily for each stockroom receipt, issue, and adjustment occasion with a corresponding computerized transaction and paper.’ This establishes a standard that states: (1) how much, (2) how it is to be done, (3) where it is to be done, and (4) when it should occur. It can be used as a yardstick in assessing the facts available to the project team about any alternative choice.

With this kind of objective, the project team can determine whether a proposed feature within an alternative does or does not satisfy the specific objective, or how closely it comes to the goal.

Objectives are derived from two general areas: (1) the results expected from the pro-

posed system design, and (2) the resources available for implementing and operating the system. The objectives that the project team sets are their guidelines for the use of these resources.

The project team members usually begin by defining what is to be expected of the system: What problems are we trying to correct? Where are we trying to go? What returns do we want for our effort? What functions are to be performed to get there? What is to be avoided, if possible? What problems do we want to minimize?

Similarly, in seeking information about resources to be committed, the project team explores the following areas:

- ☞ Manpower--skills and numbers
- ☞ Machines--equipment and facilities
- ☞ Money--recurring and non-recurring
- ☞ Materials--forms, disks, tapes, and other media
- ☞ Time--short-term, long-range
- ☞ Power--energy, transport, and authority
- ☞ Regulatory Compliance—local and global
- ☞ Interfaces to other systems—electronic data capture, document management

Using this list of resources as a guide, the project team members ask: What are the resource limits within which we must stay? What is available? What must be conserved and utilized? Where should the money be spent, and how?

The answers to these questions will provide the project team with those objectives that the selected software is expected to accomplish.

## **Step 2 – Classifying Objectives According to Importance**

All of the objectives that have been listed in Step 1 will have some degree of influence on the course of action to be selected. However, while some will be of absolute and overriding importance, others will be quite important but not mandatory, or will not affect the situation a great deal. All the objectives should be listed under two headings: ***Musts*** and ***Wants***.

The ***Musts*** set the limits that cannot be violated by any alternatives. ***Must*** objectives help the project team to recognize and screen out the impossible alternatives at the outset.

Objectives that are ***Wants*** do not set absolute limits, but express relative desirability. ***Want*** objectives are concerned with relative advantage and disadvantage.

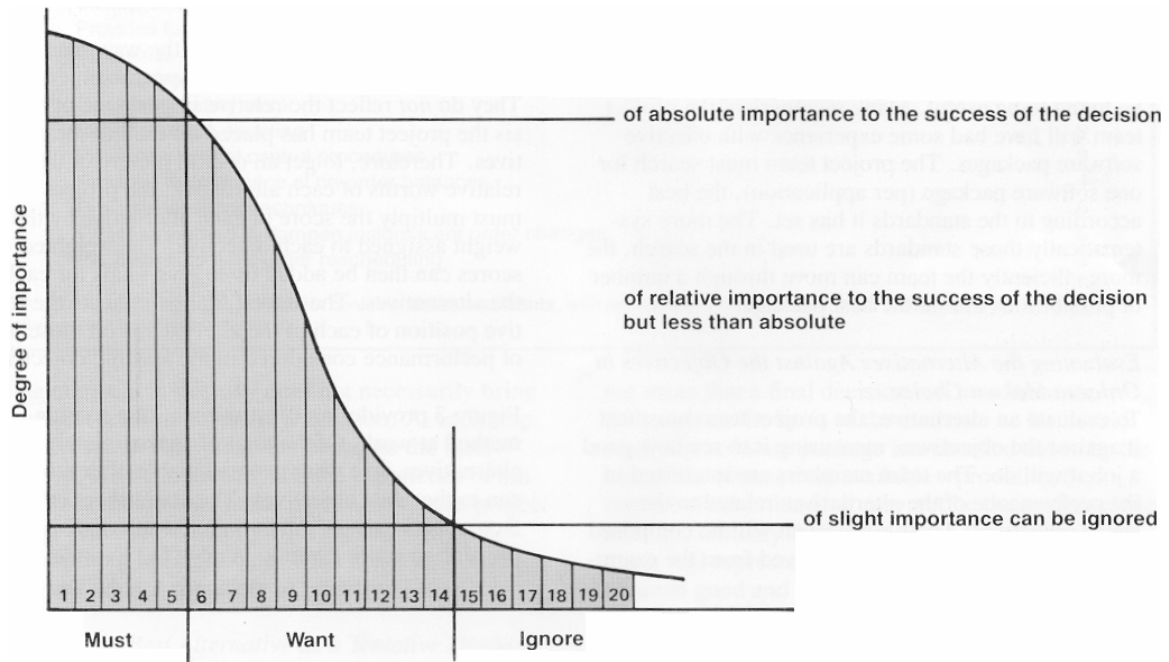
The project team will find that some ***Want*** objectives will always be more important, more critical than others. At this point, they must sharpen their judgment as to the degree of the objectives' relative importance. Figure 1 illustrates the relationship between objectives and their importance to each other. To assess this relative importance, the project team must carefully weight each ***Want*** objective.

The first step in the weighting process is to establish the position of each ***Want*** objective in relation to the next. This step is accomplished by giving each objective a numerical weight of importance.

Weighting is done by beginning with the least important objective and giving it a weight of 1. Then, it must be determined how many times more important another objective is compared to the first one. This second objective must be assigned a weight in accordance, such as 3 or 5.

Each of the remaining objectives is treated in the same manner. By this method the least important objective is taken as a standard of comparison, and all other objectives are ranged against it.

The project team's assignment of weights is drawn from its own experience. The critical point here is that some assessment of relative importance is made. The actual assignment of weights must come from the client and the potential key system users as well as from the project team.



**Figure 1 The relationship between objectives and their relative importance to each other**

Objectives considered in selecting software.

Objectives relative to the success of selection are of declining importance. Must objectives have a GO/NO-GO effect on the selection.

Want objectives carry successively less weight, and inconsequential factors can safely be ignored.

Figure 2 is an illustration of an objective-setting worksheet for a Clinical Supplies Inventory Control System software application; or a module of a complete CTMS. Many other Must and Want objectives can be added to the worksheet; the objectives shown are examples only.

<b>Must objectives: Resource limits and required results</b>	
Server CPU 2 GHz	
Operating system is Windows 2003	
Program code is Web-enabled	
On-line supported; Inquiry and maintenance	
Data base manager is used	
On-line and Off-line client	
Interfaces with company e-mail system	
<b>Want objectives: Best use of resources, maximum results and returns, minimum disadvantages</b>	
	<b>Weight</b>
Minimum capital investment	6
Lowest monthly recurring costs	8
Interfaces with company's EDC system	10
Identifies clinical supplies inventory balances by location	4
Audit trail available	7
Provides ability to handle more than one trial	9
Provides ability to handle global trials	7
Provides for independent demand ordering	5
Calculates automatically order point and quantities	4
Provides multiple-order policies	6
Calculates safety stock	5
Supports cycle counting	9
Provides physical control procedures	9
Provides maintenance of on-order balances	4
Provides lot-sizing techniques	6
Provides method to dampen insignificant order changes	5
Provides automatic order rescheduling	3

**Figure 2 Illustration of an objective-setting worksheet**

### **Step 3 – Developing Alternatives**

The set of Must and Want objectives becomes a set of specifications, systems requirements, or application criteria by which to develop a list of alternative sources. The objectives spelled out are individual statements of functions to be performed or fulfilled by the selected alternative. Once the project team has set out the functions that must be performed and has indicated how important it thinks these Wants are, relative to each other, it has the basic dimensions and requirements in hand for determining viable alternatives. This is a blueprint for what must be built. Now, all the project team has to do is fill in the details.

Finding alternatives is not a hit-or-miss affair, but a patient, careful search for specific software that meets all requirements. Reference libraries (hard or electronic) provide descriptions of vendor offerings, many of which may be found to be useful. Many members of the project team will have had some experience with effective software packages. The project team must search for one software package (per application), the best according to the standards it has set. The more systematically those standards are used in the search, the more efficiently the team can move through a number of possibilities to find the best package.

### **Step 4 – Evaluating the Alternatives Against the Objectives in Order to Make a Choice**

To evaluate an alternative, the project team must test the alternative against the objectives, measuring it to see how closely it fits. The team members are interested in the performance of the alternatives related to the stated objectives.

Their assessment will be composed of many separate judgments, each derived from the examination of many facts. To accomplish this systematically, each alternative must be measured separately against of the Must and Want objectives. First, each alternative is assessed against the Must objectives on a GO/NO GO basis.

If the alternative fails to perform what a Must objective requires, it must immediately be discarded. The Must objectives will screen out the 'impossible' objectives and reduce the number of alternatives to a relevant few. Alternatives that satisfy all of the Must objectives can then be evaluated further against the Want objectives.

To judge the performance of each alternative against the Want objectives, the project team should separately score every alternative against each of the objectives. A numerical scoring scale of 1 to 10 can be used for this provided the best alternative receives the top score. The other alternatives are then scored relative to this top score. Where there is no apparent difference between the performance of two alternatives, the same scores can be assigned.

These scores, or ranking, reflect the way that each alternative performs against the specific objectives. They do not reflect the relative performance or emphasis the project team has placed on each of these objectives. Therefore, to get an overall picture of the relative worth of each alternative, the project team must multiply the score of each alternative by the weight assigned to each objective. The weighted scores can then be added up to give totals for each of the alternatives. These total figures indicate the relative position of each of the alternatives against those of performance considered in the specific objective.

Figure 3 provides an illustration of this scoring method at work. Of the total of approximately six alternatives, five were deemed viable after comparison to the Must objectives. The score sheet in Figure 3 compares each of the five vendor alternatives, the highest score given to Vendor D.

A typical objective-setting worksheet contains Musts and Wants with appropriate weights added, as in this example, for an Inventory Control System.

Using numbers in this way does not necessarily bring the project team members to a decision. At each step, they have made their judgments based on the facts available, their experience, and the experience of others. The numbers merely make it possible for the project team to deal systematically with a great many judgments without losing track of what it is trying to accomplish and what it considers relevant.

**Figure 3 Software survey application criteria checklist**

No	Objective Criteria	Weight	Vendor									
			Vendor A		Vendor B		Vendor C		Vendor D		Vendor E	
			Rank	Score	Rank	Score	Rank	Score	Rank	Score	Rank	Score
1	Minimum capital investment	6	\$11,000		\$17,500		\$4,300		\$9,400		\$16,200	
			5	30	4	18	10	60	7	42	4	24
2	Lowest monthly recurring costs	8	\$450		\$320		\$670		\$290		\$210	
			5	40	7	56	3	24	9	72	10	80
3	Interfaces with company's EDC system	10	Yes		Yes		Yes		Yes(no pt)		Yes	
			10	100	10	100	10	100	8	80	10	100
4	Identifies clinical supplies inventory balances by location	4	No		Yes 6 loc		Yes 1 loc		Yes 18 loc		Yes Inf Loc	
			0	0	5	20	1	4	9	36	10	40
5	Audit trail available	7	Yes all xactns		Yes bal xactns		Yes bal xactns		Yes all xactns		Yes bal xactns	
			10	70	5	35	5	35	10	70	5	35
6	Provides ability to handle more than one trial	3	No		Yes 6 trials		Yes 2 trials		Yes 3 trials		Yes Inf trials	
			0	0	8	24	4	12	6	18	10	30
7	Provides ability to handle global trials	2	Yes		No		Yes		Yes		No	
			10	20	0	0	10	20	10	20	0	0
8	Provides for independent demand ordering	5	Yes thru MS		Yes		No		Yes		Yes thru MS	
			10	50	6	30	0	0	8	40	10	50
9	Calculates automatically order point and quantities	4	No		Yes (ROP)		Yes (ROP)		Yes (RP)		Yes (RP)	
			0	0	5	20	5	20	10	40	10	40
10	Provides multiple-order policies	6	Yes - 8		Yes - 5		Yes - 3		Yes - 6		Yes - 6	
			10	60	5	30	3	18	9	54	9	54
11	Calculates safety stock	5	Yes (MAD)		Yes		No		Yes		Yes (MAD)	
			10	50	5	25	0	0	5	25	10	50
12	Supports cycle counting	9	Yes		Partial		Partial		ABC Select		Partial	
			8	72	5	45	5	45	10	90	4	36
13	Provides physical control procedures	9	Yes		Yes		Batch Bal		Yes		Yes	
			7	63	5	45	10	90	5	45	4	36
14	Provides maintenance of on-order balances	4	Yes PO & MO		Yes PO		Yes Total		Yes PO		Yes MO	
			10	40	4	16	6	24	4	16	2	8
15	Provides lot-sizing techniques	6	No		Yes EOQ's		Yes 3		Yes 6		Yes 6	
			0	0	2	12	5	30	10	60	10	60
16	Provides method to dampen insignificant order changes	5	No		No		No		Yes		No	
			0	0	0	0	0	0	10	50	0	0
17	Provides automatic order rescheduling	3	No		Partial		No		Yes		Yes	
			0	0	5	15	0	0	10	30	10	30
		Total	595		491		462		788		673	

Weight = 1-Low, 10-High  
Rank = 1-Low, 10-High

Value = Quantity, dollars, etc.  
Score = Weight times Rank

### **Step 5 – Choosing the Best Alternative as the Tentative Solution**

The alternative that receives the highest score on performance against the objectives presumably is the best software package for the company. However, this does not mean that a final decision can now be made. On the basis of the scoring, this software package appears to be the best alternative (providing the resources committed). But, it may not be the best choice, only the least poor of the alternatives.

It represents the most favorable balance between good and bad as defined by the objectives set down. Most likely, the highest-scoring alternative will be a compromise between several alternatives. Each of these may be able to do part of the job better than the others, but only one alternative will rate highest in getting the whole job done.

On further consideration, this tentative decision may not be the best one for several other reasons. Whenever a new system is installed, changes to present procedures are likely to be required. Changes to procedures, whether computer procedures or manual procedures, may cause problems and delay implementation. Any software will inevitably produce effects; otherwise clients would not bother to make the decision to automate in the first place. However, since change is what produces problems, it is possible that the effects produced by the selected course of action may be worse than the initial problem.

The project team, working with the users and information systems departments, must identify these problems ahead of time, rather than wait and accept the results. The project team, therefore, assesses the consequences

of its choice as completely as it can. This is good software-selection insurance.

### **Step 6 – Assessing the Adverse Consequences of the Tentative Selection**

The project team members should take the best two or three alternatives and view them independently by visiting installations already in operation. They should question the effect each alternative would have on other aspects of the user's information systems functions, and the effect that other applications or events might have on them. They are not reconsidering the attainment of objectives, but they are estimating the 'possible' effects of the actions necessary to attain them.

For example, if lowest total cost is an objective, the team members would not consider the cost of attaining each alternative as a consequence, but would weigh the possible effects and trends of these costs over a period of time. They would ask: "If we implement this alternative, what would happen? What could go wrong?"

At this point, the project team is looking for the potential breakdowns and shortcomings that have escaped notice so far. Here are some promising places the project team might look:

- ☞ Computer Operations
- ☞ Long response times at desktops
- ☞ Documentation
- ☞ Obsolete specifications-not up to date
- ☞ Improperly designed control procedures
- ☞ Inexplicit or confusing user instructions
- ☞ Data collection forms that do not align with computer input formats
- ☞ Training
- ☞ Poorly organized training materials
- ☞ Poorly administered training course(s)

The project team can ask those sites that have already implemented the potential systems to point out problem areas in the proposed system. Key users at the other sites may provide good input at this point. A close examination of their comments could turn up some new problems that the project team might want to 'crank back' into the original set, to be weighted and treated like any other objective. The project team may also find some real dangers applicable to one software package, and to no other. Such careful examination is found to be useful in turning up hidden problems.

However, not all consequences will be equally threatening to the decision. Some may be more serious than others. Some will be fatal to the plan of action while others will merely impair it.

The consequences should be weighted in terms of severity and impact just as the Want objectives were weighted in terms of importance. In addition to severity, the project team needs to assess how probable any of these consequences might be.

After this careful evaluation, the project team may realize that because of the adverse consequences they have uncovered, they dare not select what they originally thought was the best alternative. They may find that the alternative with second-best performance of the Wants is, on the whole, the better choice because it is safer and carries fewer long-range complications.

Figure 4 illustrates the results of evaluating the adverse consequences of what the project team considered at first to be the two best alternatives.

Possible Adverse Consequences Worksheet							
Alternative D				Alternative E			
Item	Prob-ability (1-10)	Serious-ness (1-10)	PxS	Item	Prob-ability (1-10)	Serious-ness (1-10)	PxS
System does not come with validation scripts	8	4	32	User documentation is not up to the current release	6	8	48
System technical specifications not available	7	3	21	When the number of users goes past five, the response time doubles	8	6	48
				Database is difficult to interface	3	5	15
				Seven fields must be added to the CTMS	4	6	24
			53				135

**Figure 4 Possible adverse consequences worksheet**

The adverse consequences worksheet contains the possible consequences scored according to probability and severity.

**Step 7 – Controlling Effects of the Implementation of the Final Selection by Preventing Adverse Consequences and by Follow-up**

The final stage of the selection process is critical to the whole procedure. If the client and project team, ensure that they have chosen the best alternative simply give the necessary orders to implement the system, they are inviting a lot of trouble. Once the client follows the team's decision, every adverse consequence considered earlier now becomes a potential problem.

Often, implementation results fall far short of management expectations. Companies have been known to spend two, five, or more man-years to implement a major system, only to conclude that the project should be abandoned. To help avoid these pit falls, Winchester has identified several keys to successful system implementation.

## Keys to Implementation Success

---

**Define the Scope of the Systems Project:** It is necessary to specify clearly the purpose of the systems project as it relates to the client's operating objectives. A project with narrow scope may produce marginal results. On the other hand, if the scope is too wide, the system may take an abnormally long period of time to install and the implementation effort will falter through inertia or lack of interest by client personnel.

**Consider Management Goals and Company Operating Environment:** The system's design should not be based on an information systems specialist's interpretation of what runs well on a computer. Instead, the design must be based on the user's need to mitigate or satisfy his business requirements. It must cope with the dynamics of the company. The information must also be timely, correct, and appropriate.

**Establish Proper Communication Between the User and Information Systems Personnel:** Simple terminology must be developed so that the user and information systems personnel assigned to the project can readily communicate with each other. To avoid later surprises, the user must state clearly his system requirements, and the information systems department must indicate equally clearly, in user terms, how the system will meet those needs.

**Establish Responsibility for the Project:** Responsibility for project success must be defined in detail from the beginning. Most likely, this responsibility should rest with the user. The responsible party should have the authority to cut across departmental lines to keep the project on track.

**Make Management Aware of the Magnitude of Effort and Time Required:** Major systems should not be implemented without management's awareness of the magnitude of effort required to successfully complete the project. Management should be confident that the plan is realistic in terms of the resources and time required for completion.

**Provide Adequate Resources:** An implementation effort will be successful only if the company has sufficient personnel with

- ☐ technical expertise,
- ☐ experience, and
- ☐ available time to work on the project.

**Establish the Proper Operating Environment:** Company personnel must be prepared to install the appropriate disciplines in their departments in order to support a viable system. In some instances, the user must be educated in concepts and procedures in order to help personnel understand how their operations could be made more effective.

**Monitor Implementation Progress:** An adequate tracking system is required to monitor the scheduling and cost of implementation. If problems are identified, the project manager should act immediately to minimize the possible adverse impact.

The need for a usable method to assist managers in selecting the right software has given rise to the process outlined in this paper. As described here, it will lead a project team to select and ultimately implement vendor software that best fits the needs of the situation and environment.

# Winchester Business Systems

---

Winchester Business Systems is a leader in the development of cost-effective software solutions and implementation services for the clinical trials management industry. Since 1971, the company has worked on the development and deployment of systems designed to improve business process organization and efficiency. Over the past 20 years, Winchester has become one of the leading *Configured Off-the-Shelf (COTS)* software developers for the Life Sciences Industry. The company currently offers a suite of twelve complementary products targeted specifically for Clinical Trial Management.

Winchester offers its customers *turn-key* products and services that integrate seamlessly into the customer's current business processes. Winchester's basic offerings include software, consulting and customization services, documentation, and technical support.

For companies who are interested in software solutions, but clearly concerned as to whether it will meet their cost/benefit requirements, Winchester offers a comprehensive requirement study. This service is intended to assist the customer in determining its needs and perform a 'gap analysis' between its requirements and Winchester's product offering.

In the deployment of any software products Winchester also engages the clients in product tailoring and consulting services. These services are required to configure the system to meet and fit the customer's requirements, validate the system according to strict FDA regulations, train users and administrators, deploy the system, and provide initial support.

#### Winchester's COTS products include:

- ☞ Protocol Manager - CTMS
- ☞ Quality Suite:
  - Adverse Event Management
  - Product Complaint Management
  - Customer Service Systems Management
  - Corrective and Preventive Action (CAPA)
  - Training System (campus)
- ☞ Health Agency Correspondence Tracking
- ☞ paraFILE – Central Repository
- ☞ Engineering Change Control
- ☞ The Investigator Portal
- ☞ ComPac GxP Portal
- ☞ AtSignOn

# Bibliography

---

- (1) Charles H. Kepner and Benjamin B. Tregoe, *The Rational Manager* (New York: McGraw-Hill, 1965)
- (2) United States Air Force, "Parametric Evaluation of Generalized Systems, PEGS."
- (3) Herbert A. Simon, *The New Science of Management Decision* (New York: Harper & Row, 1960).
- (4) Robert J. Shaw and Michael O. Regentz, "How to Prepare Users for a New System," *Management Focus*, March-April 1980.
- (5) Michael O. Regentz, "Manufacturing Resource Planning In Now a Virtual Necessity." *Management Focus*, January-February 1979.
- (6) Michael O. Regentz and Earle F. Fay, "The Right Way to Prepare for MRP-II," *Management-Focus*, January-February 1982.
- (7) Michael O. Regentz, "Users and Information Systems: Working Together to Build and MRP System," *Computers in Manufacturing: Manufacturing Resource Planning* (Auerbach Publishers Inc., 1981)
- (8) Jack Welch with Suzy Welch, *Winning*, (Harper Business, 2005).
- (9) Michael O. Regentz, *Project Manager's Handbook*, (Winchester Business Systems, 1996)