Clinical Trial Management Systems (CTMS) System Selection Considerations
March 2004

About Author

John S. McIlwain
jmcilwain@velos.com
Velos, Inc.
2201 Walnut Ave., #208
Fremont, CA 94538
www.veloseresearch.com

John S. McIlwain is President and CEO of Velos and currently serves on management boards of several leading-edge software and healthcare-related companies.

Overview

As part of our continuing white paper series on clinical research information systems, this paper focuses on the class of research systems called "Clinical trial Management Systems," or "CTMS." In particular, we will focus on the CTMS needs and vendor requirements for research "sites" — the investigators and their support teams.

Recognition of the need for clinical research information systems has begun to move into the mainstream of the investigator market. As a result, questions relating to defining one's system requirements and differentiating vendor capabilities are emerging. The objective of this paper is to provide an informative discussion and guidelines to help readers clearly define CTMS system needs and evaluate product alternatives.

This paper is divided into three sections. In the first section, we'll address functional capability variances within the confines of a CTMS. Next, we'll look at technology issues and discuss the distinguishing considerations. Finally, we'll examine the related system capabilities that should be part of one's solution but which are not generally considered within the scope of a CTMS.

I. Capability Variances in CTMS Systems

To begin with, the term CTMS is itself misleading with respect to identifying the needs of the market. This is primarily because the term is used to identify a set of information needs that are part of a much larger set of needs. As such, most CTMS solutions are partial answers to the problems most researchers face and are therefore of some value, but not unlimited value. Despite its shortcomings, we'll use the term CTMS throughout this paper for the sake of clarity.

A CTMS solution supports the managerial and administrative aspects of running clinical trials. Most commonly, a CTMS addresses patient/subject scheduling; patient/subject status tracking; milestones, billing and invoicing; and related reporting. Some CTMS vendors also address recruiting, grant management, adverse event reporting, and some level of Institutional Review Board (IRB) functionality.

"Administrative and clinical research needs….two sides of the same coin….are interrelated."

By contrast, systems labeled Remote Data Capture (RDC), Electronic Data Capture (EDC), and Data Management Systems (DMS) generally support clinical information needs. In our view, administrative and clinical research needs are interrelated. Some CTMS vendors focus on the needs of study sponsors; others focus on the needs of investigators. Within the investigator CTMS market, this paper addresses variances in how CTMS vendors manage the above range of possible needs.

In many cases, there is great value in solving the limited set of problems most CTMS address. For example, most large research programs using a good CTMS can justify their investment as a result of improved research service reimbursement alone — usually inside a year or two. However, most CTMS fall short in their ability to extend and facilitate interaction within the organization (e.g., integration with internal information systems), with sponsors (e.g., through electronic data capture), and with research collaborators (e.g., through multi-organization support). Without these system extensions, CTMS solutions can perpetuate a fair amount of work duplication rather than reduce or eliminate it.

I.a. The "Features" versus "Capability" Challenge

A Request for Proposal, or RFP, generally contains a long list of required and desired system features. Two vendors can mark "Yes" to 85% of the features but have different capabilities. To illustrate this, in the business accounting software market some vendors have products that sell for less than $1,000, others for over $1,000,000. In a typical RFP for a business accounting system, both vendors can usually mark 85% "Yes." While both products have similar features, they have vastly different capabilities. While we do not see the same degree of variance in the CTMS market, it's not just that the box is marked "Yes" that is important; it's how the vendor does it that makes the difference. It's these variances that we'll explore below in greater depth.

> "It's not just that the box is marked "Yes" that is important; it's how the vendor does it that makes the difference"

I.a.1. Configurability

One of the first items to be considered for long-term CTMS success is configurability. Be sure the vendor's product can be adapted to your needs as your needs evolve. Clinical research is in a constant state of change, including regulatory changes, work volume, work allocation and related job responsibilities.

What you are probably looking for is a system that meets your current requirements. What you really need is a system that meets your immediate needs as well as your future requirements, preferably with minimal programming changes or other forms of vendor reliance. It is relatively easy for a software engineering team to meet a specific set of functional requirements for a given point in time. A system that is designed to evolve with your needs, without software code changes, is far more difficult to accomplish — and has far greater value for you because you won't have to think about changing systems in a few years when your needs change.

I.a.2. Interoperability

Webster defines "interoperable" as "capable of being used or operated reciprocally." The idea of interoperability in software is simply that an action taken, such as the input of data, in one part of a system impacts the other parts of the system, or systems, that have a relationship to that action.

> "(Event) interdependency – your system should take care of this…."

Why do you need a system with strong interoperability? Good question! A simple example of the benefit of interoperability is upon completion of a patient visit. At this time, a number of events can or should happen. Results are recorded – if the results are in one expected range, certain persons must be notified or action taken, if in another a different set of actions and persons are involved. Each such individual or group usually requires different information. The system should take care of these interdependencies for you.

The completed patient visit could trigger a billing event. One would rather not have to re-enter information into a separate billing module since most of data required to bill should already have been captured. A sponsor might be looking for particular supporting documentation for the invoice. Your clinical trials office might need other financial information, perhaps in a different format. The investigator may want the same information but in another format and at a summary level.

Then there are new events to be scheduled for the patient as a result of the completion of this event: new labs need to be ordered; a special reminder is required; a room needs to be reserved. And so forth. Your system should be able

to complete all these tasks without data entry duplication and without human intervention beyond entry of the usual information required to complete a patient visit.

So how can you get all this in one system? Look for a system designed to be interoperable. Currently, most systems are developed using a functional approach. A function is a specific type of activity: scheduling, billing, reporting, or emailing. In a functional approach to software development, a complete functional module is developed independently, then it is determined how to make the functions communicate. This is relatively easy for software development teams to envision and develop, but results in systems with very limited interoperability.

> "Software that handles interoperability well is an essential system selection consideration."

By contrast, a software team focused on interoperability, using modern object-oriented techniques (discussed briefly below), creates "classes" of functions, which can be given different "attributes," depending on the particular level of use, which can be "instantiated" (or inserted) where needed in the application. The result: a system that is far more interoperable and configurable.

So, in selecting a product, understand how the vendor developed their product, not just what functions it is capable of fulfilling. Since the vast majority of CTMS were developed using a functional approach, vendors with products that have been on the market for awhile have often built many of the software bridges required for internal software functions to interoperate. As such, the "Yes" column in the RFP is checked and an immediate requirement may be addressed.

However, such bridge building gets more and more complex for each additional feature, and the software becomes more and more difficult to change. The software improvements come slower and slower, and eventually product evolution grinds to a halt. At this point, the vendor must develop a new system and the customer has to change systems — or change vendors.

Clinical research is evolving rapidly. Software that handles interoperability well is an essential system selection consideration.

(Note: For an in-depth discussion of interoperability in clinical research, please see the Velos paper entitled "The Way to High Returns on Systems for Clinical Trials").

I.a.3. Workflow Orientation

Workflow orientation, like interoperability, is a departure from a functional approach. A workflow-oriented system mirrors the work activities of the individuals engaged in the research process. The above example of the patient completion event shows that one individual, such as a study coordinator, "touches" many functional aspects of clinical research. The result of a workflow-oriented system is that it is much easier to use, requires relatively little training, and helps prevent entry mistakes.

The best model for supporting workflow in clinical trials is actually the same that might apply to patient care. This is clearly demonstrated in the chart below, taken from the June 2002 Velos paper."

Name of Patient Care Module	Name of Clinical Research Module
Treatment or Careplan Design	Protocol Design
Patient Scheduling	Study Subject Scheduling
Electronic Medical Record	Case Report Form
Clinical Compliance System	Query Management System
HIPAA-Compliant Security and Audit trail	HIPAA-Compliant Security and Audit trail & 21 CFR Part 11
Data Interface Engine	Data Interface Engine

JAHCO and other Compliance Reporting	IRB and other Compliance Reporting
Budgeting and Billing	Budgeting and Billing

The same principals applied in patient care are applicable to clinical research. It follows that the investigator market is better served by a vendor with a patient care system background as compared to a purely research administration or a sponsor background. Clinical research in many respects is a subset of patient care. A vendor with extensive provider-side system experience probably has the best understanding of your workflow needs.

How easy is the system to use? Is the system intuitive for end users? If so, this reduces training costs and fosters far higher system utility, translating to higher productivity and staff satisfaction. The return on investment from your new system is directly proportional to the system's ability to match your current and future workflow.

I.a.4. Communications and Messaging (Work Groups)

Let's move to work groups. Clinical research is a team activity that involves good communications within the research team, the organization, and with third parties such as patients and sponsors. Today, most communications occur in relatively non-systemic form: phone calls, email, staff/conference meetings, and in passing. In clinical research, the reality is that much of this communication is systemic in nature. Specific events trigger the need to issue or receive specific communications. Also, any sort of best practices or quality assurance programs can be made far more effective with systems-based messaging and communications.

> "A product that is configurable, interoperable, and workflow-oriented … is valuable (to you)."

Having a product that is configurable, interoperable, and workflow-oriented, with a complete messaging and communications infrastructure within your product, is immensely valuable — and it's challenging for vendors to implement well.

I.b. Risk of Oversimplification / The Single Perspective Challenge

It's important to identify the priority problems you would like the system to address first and to be sure the vendor can address these needs. Usually, a vendor cannot be "everything to everyone." However, particularly in clinical research administration, there can be a tendency to oversimplify the problem.

> "To improve reimbursement, the key system challenge is … supporting the individuals who interact with the patient."

For example, many organizations we meet are interested in purchasing a CTMS product to increase their research reimbursement. In most cases, this is because some services are not routinely being captured and billed. If a priority is to be sure that research services are invoiced, the best way to do this is to be sure the billing data is already captured as part of the patient administration process. A system's effectiveness in supporting patient administration is in turn dependent on its ability to help coordinators, who have a different set of tasks and concerns. If the system is being used for patient administration, a well-designed system will essentially generate the bill. To improve reimbursement, the key system challenge is not generating the invoices and bills; it's supporting the individuals who interact with the patient.

This is similar to clinical information systems for patient care. If the system does not help the people at the point of care, it's going to be more difficult to achieve the financial and other administrative results you're looking for. This is the inverse of what you might expect a CTMS system to focus on — but the only way you'll achieve improvements. So, set your priorities, but be sure you support all the human perspectives that will enable you to accomplish them.

Look for vendors that can support multiple human perspectives, particularly at the patient interaction level. Find out if the vendor's product is most commonly used at the patient interaction level.

> "Is…the vendor's product most commonly used at the patient interaction level?"

I.c. What About Price and Quality?

The software industry is far from a commodity business. There are great variations in software quality that directly impact productivity, staff satisfaction and return on investment.

In clinical research administration, there is a high degree of variability in the return you'll get from your system for two reasons: 1) there's a great deal to gain, and 2) once implemented, the level of actual system usage varies greatly. The return on investment from successful system adoption is so high that, as a rule, the better system will produce far better returns that the mediocre system and easily justify reasonable price differences. Price is an important consideration, but it should not be the primary one.

> "Price is an important consideration, but it should not be the primary one."

Another price-related consideration is total investment and implementation costs. In a 2003 study sponsored by the Association of American Medical Colleges, entitled "Information Technology Enabling Clinical Research," the authors estimated that for every $1 spent on software and hardware, healthcare organizations spend $2 to $10 on implementation! Different software products will contribute to different levels of implementation and support costs, or savings. These indirect costs/savings are probably more significant than the actual cost of the software. Understanding these costs, and a vendor's ability to help control them, is probably as important as the price paid for the software.

> "Consider total investment and implementation costs."

One of the advantages of some of the newer, Internet-based software technologies is that vendors who have developed products in this technology have been able to develop better software at a lower cost. Choosing such a vendor can result in a better product, with a far longer useful life, that is easier to use and less expensive to maintain — and might not even cost much more than the alternatives.

> "…better software at a lower cost."

II. Technology Considerations

From capability differences in CTMS offerings, let's move to technology selection considerations, where there is also a great degree of variability. We'll limit the terminology in this discussion which we hope will be informative for managers, end users, and technologists.

Internet vs. Client-Server: On the technology side, the biggest area of differentiation among CTMS systems has to do with Internet versus client-server technology. (Encompassed in the term Internet, as used here, is the class of technology that also includes intranets and extranets). Virtually all CTMS products today are accessible via the Internet. Any client-server product is accessible on the Internet — therefore any vendor can claim Internet capabilities.

Most systems that support clinical research were designed and developed in client-server technology that was current 6 to 12 years ago. It is expensive, and risky, for software vendors to switch system designs and infrastructures, and introduce "discontinuous" improvement in their product line. Instead, such systems are accessible via the Internet using a "wrapper" technology — a third party software tool that a vendor can purchase and "wrap" around its product so that it can be accessed on the Internet. If you want to avoid a discontinuous technology shift far sooner than necessary, what you should look for is most likely a true Internet product.

> "Few (systems) are designed for the Internet."

Top

The Smaller "Footprint"

In software technology, the "footprint" is the impact a software product has on one's systems infrastructure – generally, the smaller the better. Internet technology enables organizations and users to obtain selected access to information and systems, which results in better communication and more efficient distribution of work, using a

smaller footprint to do it. The cost of maintaining, updating, and distributing systems developed for the Internet is significantly lower than client-server platforms.

XML

Velos uses XML, which stands for eXtensible Mark-up Language and is the successor to HTML – or Hypertext Mark-up Language – the foundation programming language for the Internet. Essentially, XML enables Internet software developers to pre-define relationships between data elements so that when the data elements are used again for similar purposes, the relationship does not need to be defined again.

Systems developed using XML can be adapted to support new or changing information requirements faster and at a lower cost than systems using HTML or SQL. This helps prevent the need to rewrite software applications, or portions of software applications. XML "extends the life" of a software product. With XML, the same, or similar, information sets can be presented in different formats to different end users without writing new software.

Object-Oriented Software

Object-oriented software design has been around for quite awhile, but generally has not been incorporated in most software products. The advantages of object-oriented software are a bit like XML except that it works at the whole application level as compared to the reporting, or data, level.

By way of example, a type of object, or an object "class," might be a patient. A patient has certain attributes: a name, a diagnosis, and a birth date. A patient appears in many different instances, or functions, of a product, such as scheduling, billing, and reporting. In object-oriented software, a class, called "patient," is created once and "instantiated" in the form of an "object" wherever it is needed in the application. Variations in the "attributes" of the class "patient," required for different functions, are addressed by "turning on or off" attributes for any particular instance.

Why do you want object-oriented software? In short, the useful life of a product can be significantly extended, which in turn dramatically improves your return on investment. Object-oriented software generally takes longer to build initially because it is important to think through one's classes. It's more expensive to build and requires a more sophisticated engineering team. Once this work is done, changing the application is much easier; the application is more reliable because fewer software code changes are required as one's needs evolve; and, therefore, more changes can be made without effectively changing the core structure of the application or affecting a "rewrite."

The difference between object-oriented software and traditional software is substantial. Very few CTMS products are object-oriented.

HL7 RIMS

In addition to the above benefits of XML and Object-Oriented software, there are other specific benefits that tie back to future technology innovation. One area of innovation specific to healthcare has been the development and adoption of HL7 (""Health Level 7"). HL7 is a set of data definition standards that enables healthcare systems and products to exchange data more effectively. The HL7 standards, which have been widely adopted, have vastly reduced the amount of work required to enable related systems to exchange data.

The next generation of HL7, which is now in use, is called HL7 version 3, or HL7 RIMS. RIMS stands for "Reference Information Models." The goal of HL7 RIMS is to enable better testing of a system's conformance to a set of information standards and to increase the utility of data exchanged between systems.

HL7 RIMS is entirely object-oriented and XML-based. As a result, it's quite important for your vendor to use object-oriented and XML technologies in their products if they're going to enable you to benefit from the new HL7 technology. Taking advantage of these improved data communication standards will require a fundamental rewrite of most client-server based products.

"HL7… enables healthcare systems and products to exchange data more effectively."

Component Software

Among our customers, one of the requirements we're seeing today is the need to make available parts of the system to some users, and other parts to other users. To accomplish this requires the software product to be available in components. In traditional software product design, some such access can be allocated through passwords. However, every time limited-need users access the system, they exert the same level of pressure on the system infrastructure, as does a full-use end-user. In contrast, for example, a Yahoo can support millions of hits on their website because their software was developed in components which only taxes system resources to the extent required.

Is the software component-based? How do you control data access? You need to look for a component-based software product. It would very difficult to build component-based software without first committing to a full object-oriented approach to system design and development. As such, very few CTMS systems are available in components. (Note: Licensing a module is an entirely different matter.)

Top

III. CTMS-Related System Needs That Warrant Consideration

So far, we've identified significant functional and technology differences among Clinical Trial Management Systems within the scope of how CTMS solutions are currently defined. What about the other side of the coin? What about related needs in areas of Electronic Data Capture, Adverse Events, system-based recruitment, and integration with your clinical systems? What about the entire clinical domain? What about supporting research collaboration?

To be sure we all understand why the clinical side is important, let's go back to the patient care perspective. As noted above, the activities of patient care and clinical research are nearly identical at a conceptual level. In one, the patient goes through a treatment protocol, the other a research protocol. Whether for research or routine care, patients are scheduled, visits are recorded, labs are processed, services are invoiced, and regulations are complied with.

"…a CTMS is highly related to clinical research and both are related to patient care."

In an ideal world, the same system that supports patient care, the electronic medical records, or "EMRs," would support clinical research. Clinical research would be a subsystem of patient care, and research administration would be a subsystem integrated with the clinical side of research. We don't live in the ideal world, but we are moving in that direction. In other words, a CTMS is highly related to clinical research and both are related to patient care. Relatively soon after you acquire your CTMS system, you're going to be faced with the question of how to integrate it with the clinical side in both research and patient care. With this in mind, below are a few CTMS-related competencies you might consider in your choice of vendor.

"The question will arise: How do I integrate a new CTMS with the clinical side in both research and patient care?"

Electronic Data Capture

Since "Electronic Data Capture" is, as we've described above, the other side of the CTMS coin, it would be great if you could have Electronic Data Capture properly addressed by the same vendor. This way, all the points of integration between the two systems, or subsystems, would already be in place. What's more, the coordinators and other "hands-on" professionals, the people you need to have use your CTMS, can accomplish more by addressing clinical data needs at the same time. Even if you decide to not address Electronic Data Capture now, or if you already have an Electronic Data Capture product, your CTMS vendor should be able to support integration well. Obviously, a

CTMS vendor who also provides Electronic Data Capture is in the best position to support such integration, because they've already built the integrated application. Very few CTMS vendors also do Electronic Data Capture well.

> "… your CTMS vendor should be able to support integration well."

System Integration and Interfaces

The next area to consider is system integration. A CTMS system might have the capability to support subject recruitment and allow you to create a set of exclusion and inclusion criteria for patients. However, unless one has the ability to electronically populate databases from which to screen patients, you are likely to miss suitable research opportunities for your patients. Or, if you manually create such a clinical database, you have to do a lot of work to manually update it. This manual activity is what some more systems-savvy investigators are reluctantly doing today. Another area of integration is labs and adverse events. One of the main reasons to have a study management system can be to facilitate management of adverse events. Like patient screening, adverse events can also be managed manually. Indeed, depending upon the state of the lab systems you now use, you may have no choice. However, you're likely to want to automate adverse events monitoring, now or in the relatively near future. The technology is certainly available today and the economic argument for automating adverse events is generally quite compelling.

> "Look for Vendor experience in system integration and interfaces.

If the CTMS vendor does not have extensive system integration and interface experience, it's going to be very difficult to automate either of the above applications. The same, of course, applies to electronically populating the case report forms with clinical data such as labs. It may be possible to hire an additional party to address interfaces (in addition to your vendor(s) and your internal systems organization), but that comes with additional costs and headaches. Vendors who have experience providing interfaces as part of their service generally have existing libraries or working interfaces which significantly reduces the total investment and implementation time. Thus, the cost of ownership goes down substantially because manual labor associated with routine system use can be significantly reduced.

Support for Research Collaboration

Apart from the fact that most research organizations still primarily use paper, one of the biggest opportunities for clinical research automation is to enable research collaboration to occur electronically. By collaboration, we do not simply mean Remote Data Capture (RDC) as has become more common among research sponsors — as deployed by sponsors today, RDC often involves more work, not less, on the part of the site.

By collaboration, we mean the ability for a single system to support multiple studies and multiple organizations within a single system framework. A system that supports such collaboration has capabilities in the areas of security, system hierarchy, and functionality that are an order of magnitude apart from most CTMS systems on the market today. Once you deploy a CTMS system to meet your internal requirements, research collaboration is likely to be an area you'll want to address next.

Conclusion

While there are similarities across CTMS systems as measured by functional checklists of features, there are significant differences in how systems are designed. In functionality, these variables include significant differences in configurability, interoperability, workflow, and messaging and communications. In technology, most notable are the advantages in systems designed for Internet platforms versus client-server platforms. In the longer term, these differences have a particular economic impact.

Remember, CTMS is a label. It circumscribes a set of functionality that is part of a larger requirement set, including clinical aspects of research and related integration requirements. While it may not be necessary to address both requirements at once, nor through one vendor, it is important to recognize that CTMS is one side of the coin. A vendor who can address all your information needs, and both sides of the coin (administrative and clinical), will better serve your long-term interests.

Category	Specifications

I. System Capabilities

General	Have I differentiated between system "features" and system "capabilities"?
Configurability	How will the system support my needs as they change?
	Was the system designed to be configurable without additional software programming or vendor intervention? How?
	What specific software design techniques were employed to support configurability? (requires a technical evaluator)
	Is the system object-oriented? Has the vendor explained / demonstrated how? (requires a technical evaluator)
	Do I understand the value of object-oriented? Does the vendor understand what it means to be object-oriented?
	Does the system support user-defined data dictionaries?
	Does the system enable the user to set up its own libraries for capabilities such as patient encounters, schedules, case report forms, and reports?
	Can these be changed easily by end users?
	Does the vendor provide tools so that the customer can add system functionality non-programmatically and without vendor intervention?
Interoperability	Was the system designed with a focus on functions or interoperability?Ask the vendor to demonstrate inoperability by providing an example of where you need functions to "communicate."
	A good one is the completion of a patient event. Is the information entered reused everywhere it's required?
	What if your interoperability needs changes . . . How does the system handle that?
	To what extent will "one click" result in all related information needs being addressed? For example, if the study coordinators use the system in the clinical setting, can I be certain that all the invoices all the billing data will be captured? Will the adverse events be properly recorded and reported? Will the investigators get the information they want?
	If I make a change to how the system is set up in one part of the system, is that change reflected throughout the system?
Workflow Orientation	Does the system mirror the way we work? Does it consider each type of end user and their workflow or information needs?
	Can the system be set up to mirror the way particular individuals or groups of individuals work?
	Does the system store these methods so they can be reused at will?
	Is the system natural and initiative to use? Can I change the workflow?
	Was the system developed primarily from a patient care workflow perspective or from an administrative or "back office" workflow perspective? Will the study coordinators and nurse managers like the system?
Communications and Messaging	Does the system have built-in, user configurable communications and messaging?
	Are these messaging/communication features available everywhere in the system that it may be required? Is the look and feel the same?
	Is the communications/messaging context-sensitive? Conditional? For example: If event A happens, can message B is triggered? Do I have complete control over how the messaging/ communications are set up?
Price/Quality	How am I going to get the greatest returns from the system?
	Who needs to use the system to be sure I get that return?
	What's the return on investment if I can get clinicians to use the system as compared to simply using the system as a secondary "back office" tool?
	Which system is most likely to be useful to the clinicians?

Which system best reduces tasks that are redundant today?

Which system can prevent mistakes, ensure required communications, enforce regulatory compliance, and foster study compliance?

II. Technology Considerations

True Internet Platform	Do I understand why true Internet technology is important?
	Was the system built with true Internet technology or an Internet "wrapper"? (Hint: if the programming language used to build the system is not Java or Microsoft.Net, the system is very likely NOT built with Internet technology.)
	Does the system support research consortiums as well as in-house researchers?
The Small Footprint	Does the system require hardware or network capability that is in any way additional to what is required to run any Internet application, such as Yahoo?
	In particular, do I need additional processing power for end user workstations?
	When software updates come, do I need to make any changes to end user workstations, or are updates reflected across the network from one server? If the latter, do I need to buy any special equipment to enable such network-wide updates?
XML	Do I or the vendor understand what XML is and how it's different from SQL or HTML?
	How does the vendor's system leverage XML? (Merely Exporting data in XML-compatible formats does not leverage XML.)
	Do I or the vendor know what HL/7 RIMS is? How can I be comfortable that the vendor will be compatible with HL/7 RIMS when it becomes a standard.
	Does the vendor even have any clinical data integration capabilities?
Object-oriented	Do I understand the value of object-oriented software? Does the vendor? Can he/she explain it to you?
	Is the system object-oriented? If so, can the vendor explain/demonstrate how it uses object classes, inheritance, and object instances (requires a technical evaluator)?
	Was the system developed in components? Can a single component be purchased or enabled for use without the rest of the system? Can the user allocate system components to internal and/or external end-users without vendor intervention?

III. CTMS-Related Needs

Electronic Data Capture	Does the system provide a complete, integrated clinical research information system?
System Integration and System Interfaces	Does the vendor have an Interface Engine?
	Does the vendor have a background in provider-side clinical information systems? Do they also offer a patient care product line?
	Does the vendor have proven experience providing clinical system integration?