Avoiding the Pitfalls in Information Systems Contracting

By: Cathy Kiselyak Austin

DrinkerBiddle

I. INTRODUCTION

Drafting and negotiating contracts effectively requires skill and knowledge with respect to both standard contractual issues (*e.g.*, confidentiality, notice, etc.) and those specific to the subject-matter of the transaction being negotiated. Information systems transactions, whether involving standard software and hardware or Internet-related systems, raise a number of distinctive practical and legal considerations.

This article discusses several key issues characteristic of information (IS) systems transactions and some of the strategies that can be used to address them successfully.

II. IS CONTRACTING HAZARDS

Most companies engaging in IS transactions are users, rather than vendors. From the user's perspective, the primary purpose of the governing agreement is to ensure that the user will receive what it is paying for. This is particularly difficult to achieve for IS projects, because they tend to involve new technologies and require the completion of complex integration and implementation tasks on tight deadlines. Some of the most challenging aspects of IS contracting are:

- Structuring the project to succeed despite short time-frames and moving technology targets.
- Setting performance standards that can be understood and enforced.
- Securing appropriate ownership rights in customized or newly developed software created by a service provider.
- Using payment terms as incentives for timely performance.
- Making sure the *entire* system functions effectively in a production environment prior to final acceptance and payment.
- Securing adequate support and maintenance for the post-launch period.
- Determining when source code escrow is necessary and securing adequate escrow rights.
- Designing and implementing the technical and administrative aspects of the system to comply with regulatory requirements (*e.g.*, privacy, security).

In providing legal counsel for an IS project, it is essential to be aware of key issues early in the process. In a Web site development project, for example, the structure of the site and the nature of its interaction with databases and other software may be determined in part by regulatory requirements. Similar to the way tax concerns drive the structure of many corporate deals, privacy and security issues can significantly impact the design of any system that handles sensitive information.

In addition, it is important to structure the contract to enhance its long-term effectiveness in the short-term environment characteristic of many IS projects, which usually don't have long lead times and, in some cases, may not have the benefit of managers experienced in complex technology development or implementation projects.

Section III below provides an overview of some of the key IS contracting issues and strategies to resolve them.

III. IS CONTRACTING STRATEGIES

A. *Design the Agreement to Meet the Challenges Inherent in IS Projects.*

Most IS projects are intense and depend on highly motivated, mission-oriented teams who work long hours to get the job done. The governing agreement should both encourage and reflect that. To use a military analogy: good fighter plane designers assume that the planes will be shot at and design them to help the crew survive, complete their mission, and return home intact.

An attorney who is aware of the business and legal issues and armed with the right tools can build safeguards and flexibility into an IS agreement that will facilitate both the initial negotiations and the resolution of the inevitable problems that the project will encounter. The power of a good agreement stems not so much from its creation of legal rights as on its incorporation of good project management techniques that require the parties to articulate their expectations clearly and utilize effective change management and problem resolution techniques.

The primary characteristics of a typical IS project are speed and a desire for superior technology. The short product life-cycles in this industry provide an incentive for managers to purchase cutting-edge technologies in order to minimize the possibility that the technology will be obsolete before the project's completion. These factors complicate both the management of IS projects and the development of an effective agreement.

The attorney responsible for the transaction usually faces the following challenges:

1. **Legal Structure**. Pre-project planning is typically in a rudimentary state at the time the project begins and the main agreement is to be drafted. Often the technical staff is unable to articulate exactly what they want from the vendor. The attorney has to work closely with both the client and the vendor to identify the specific goals of the project, describe them in terms that facilitate dispute resolution and determine what legal provisions best protect the client's interests.

2. **Exit Strategies and Payment Terms**. For projects with aggressive deadlines, usually neither the business people nor the vendor feel that they have the time to develop the high quality, detailed specifications that a good IS agreement requires. Paradoxically, however, the more aggressive the project time frame, the greater the potential negative consequences of poor performance and project delays. The role played by good exit strategies and well-structured payment terms is correspondingly more important and more likely to be a factor in project success.

3. **Vendor Responsibility for Technical Design and Implementation**. As the "hired expert," the vendor/implementer should be legally responsible for ensuring that the technical elements are selected, developed and installed in such a way that the finished system meets the specified performance metrics. If the client is unfamiliar with the technology, the acceptance criteria should focus on performance metrics that the client can use and understand (*e.g.*, availability, responsiveness, etc.) or can be readily validated.

4. **Project Management**. The importance of flexible project management and regular communication between the client and the vendor should be addressed early and built into the agreement. Later, during the development, implementation and acceptance phases, this will be critical to the progress of the project and the client's sense of satisfaction with the final result.

In short, a good IS development agreement must be flexible, assume that changes (and problems) will occur throughout the project, give the parties sufficient tools and incentives to resolve routine problems, and protect the client in the event of catastrophes.

B. *Use Legal Terms and Project Structure to Facilitate Early Problem Resolution.*

Complex projects always encounter problems. The more clearly the parties' obligations are defined, the easier it is to determine who is responsible for problems and avoid disputes. The more effectively the problems are handled, the less likely they are to result in cost overruns and litigation. Problems are most easily resolved if the agreement provides:

- Objective standards of performance.
- A clear definition of each party's role and responsibilities.
- A project management structure that requires regular communication, formal issue tracking, and cooperation in problem resolution.
- A dispute escalation process that allows the project leaders to pass serious disputes up the management chain before external forums may be utilized.

C. *Keep the Pressure on for Clear Specifications.*

In IS projects, there is a tendency to sign the contract and begin the project before the specifications have been formulated. This should not be an excuse for the vendor to avoid standards. It is essential for the target of the project to be defined well enough to give the parties an objective standard to determine when the vender has met

its obligations. Good specifications minimize disputes. Their usefulness as a gauge of contractual performance is directly related to how precisely they are defined.

If the project is on a very fast track, one way to advance the project is to first agree on preliminary specifications with clear overall performance standards (such as functionality, ease of use, speed, security, etc.), and incorporate a project structure that requires the development of final specifications, a detailed project schedule and a responsibility list within a short timeframe, such as two to four weeks. This allows the primary vendor to be involved in the development of the specifications and take responsibility for them.

Negotiating those specifications is often tedious, but a critical protection for the client and essential for effective project management. Information systems are complex and typically do not function satisfactorily unless all of the elements function well together. This increases both the importance of contractually requiring the development effort to meet detailed specifications and the difficulty of formulating them.

If it proves to be impossible to agree on the specifications at the project price, the client should be able to walk away from the deal.

If the vendor insists that it is too early in the project to establish specifications or a project price, the attorney may want to suggest a short initial consulting engagement during which the vendor would do any necessary research and work with the client to develop such specifications, together with a budget and schedule of deliverables.

Be aware of the risks involved in allowing either the schedule or the specifications to be developed after execution of the agreement. If the attorney does not stay involved with the project, the schedule and specifications agreed upon by the parties may not adequately protect the client's interests. In the worst case scenario, the parties never draft the schedule or the specifications and instead rely on undocumented meetings and conversations to manage the project. It then becomes extremely difficult to terminate the vendor for poor performance, as the client must first prove what the applicable standards were and has little evidence to support its position.

D. *Address Ownership and Licensing Issues.*

A less obvious, but equally important, issue that must be addressed in the IS agreement is allocation of the ownership and usage rights for the deliverables. Typically, some of the deliverables will be purchased or licensed from the vendor or third parties, while others will be contributed by the client or developed specifically for the project, such as custom software modules or Web site graphics.

Many clients do not understand what their rights are with respect to these deliverables, particularly with respect to materials developed for them by the vendor. One of the tasks all attorneys face in counseling IS clients is explaining these rights to the client, determining what their needs are and developing contractual language that meets their needs in a manner acceptable to both parties.

Under the U.S. copyright laws, unless the agreement specifically provides otherwise, the client will not own any part of the final product, except those elements it already owns, such as its logo or code drafted by an employee. The fact that the deliverables may have been developed expressly for the client and at the client's expense does not affect the issue of ownership. Without appropriate contractual protections, the client will have only a nonexclusive license to use the deliverables, without any right to prevent the vendor from copying them and licensing them to others.

If the client requires exclusive use of the deliverables, either an exclusive license or a transfer of ownership is required. Most software does not qualify as a "work made for hire" under the U.S. copyright law, so if the client desires to own it, the vendor must explicitly assign its ownership rights to the client.

Generally we recommend that deliverables developed specifically for the project belong to the client, unless they are generic in nature and the vendor insists on being able to use them in future projects. In that case, the agreement could provide either that the client owns the deliverables and licenses them to the vendor or that the vendor owns the deliverables and licenses them to the client. Either way, if the client is to pay the full cost of their development, the client should consider requiring the vendor to share any revenue that the vendor may realize from their use by third parties.

The use of preexisting materials owned by the vendor may make the project less expensive, but shouldn't be allowed to limit the client's use of the system. All preexisting vendor materials that are included in the deliverables should be identified by the vendor and licensed to the client for an appropriate period of time, possibly

in perpetuity. If they are developed by a third party, but selected by the vendor, the vendor should take some responsibility for their functionality and integration.

Further, if the system requires tools proprietary to the vendor for optimal use or maintenance, the client should consider securing a right to license such tools. For example, if a Web site developer designs a site for the client that cannot be updated without using the developer's proprietary software tools, the client should either license the tools at the outset of the project or secure an option to license them in the future. Otherwise the client will be locked into using that developer for site maintenance. If the maintenance services become too expensive or deteriorate in quality, the client may have to abandon the site. Another solution to this problem is to include in the agreement a provision that grants the client a ceiling on any price increases and a royalty-free license to use the software tools in the event of breach of the maintenance agreement.

E. *Structure Termination Provisions Consistent with Project Deadlines*.

If the scheduled time to launch the system is short, poor performance cannot be tolerated for long without adversely impacting the project. Changing vendors inevitably delays the project's completion yet further. To reduce the length of these delays, make sure that:

1. The testing and correction periods for interim deliverables are appropriately short; one to three business days is not unusual for projects that are on a fast track.

2. The cure period for material breach of the agreement is consistent with the project's schedule (e.g., five or ten days, rather than thirty, for fast-tracked projects).

3. The vendor warrants that the deliverables will conform to the specifications and the specifications are well-defined.

4. Whenever possible, the client has the right to terminate the agreement at any time, for any reason. This type of "termination for convenience" clause is more palatable to the vendor if accompanied by a requirement that the vendor's fees continue to be paid for a period of thirty or sixty days to allow the vendor time to reassign its personnel.

In addition, to prevent the vendor from being able to hold the project hostage, minimize the amount invested in unrealized assets at any point in the project. This requires:

1. Scheduled delivery of interim deliverables to the client in a form that would allow them to be used as a starting point by another vendor. This can be encouraged by making acceptance of such deliverables a prerequisite for interim payments.

2. Delivery of all work in progress promptly after termination.

3. That the client have full ownership of all original material included in interim deliverables and work in progress and an adequate license for all preexisting material.

F. *Acceptance Issues*.

A typical IS project has several distinct phases – planning, design, development, implementation, testing and acceptance, and launch – which in a complex project may occur simultaneously for different portions of the project. The acceptance phase is the client's last opportunity to ensure that the final outcome of the project will meet its needs. It should be carefully structured to enable the client to provide meaningful review of each deliverable and allow adequate time for the correction of any problems.

Well-drafted acceptance provisions can be critical to the success of an IS project. They require:

- Clear and meaningful acceptance standards.
- Well-timed testing, correction and review requirements.
- A comprehensive final acceptance requirement.

The acceptance standards usually have to be drafted by the attorney based on discussions with the client's technical staff. A good way to start these discussions is with an initial draft derived from the vendor's claims concerning the product's capabilities, as described in the vendor's literature and Web site. (For information on the performance levels of customized or newly developed products, check the original development agreement, statement of work or response to RFP). This helps the staff to focus on what they need the product to do, what level of performance would be unacceptable and what types of failures would be catastrophic.

The timing of the acceptance testing, correction and review requirements are critical to the effectiveness of the exit strategies built into the IS agreement (see above). They must be carefully drafted to prevent slippage in the project schedule. The vendor should not be able to use them to draw out its performance significantly beyond

the original schedule without triggering the client's right of termination for breach.

The most neglected aspect of acceptance typically is final acceptance of the entire, integrated system. The success of any information systems project depends on the final integration of all of the individual pieces within the client's broader IS environment. The ability of those pieces to perform smoothly as a single system within the production environment is essential, yet difficult to assess until they are tested together.

Built into the project schedule should be sufficient time to have the system tested after it has been integrated with all of the systems that are intended to support it or communicate with it. Even after such testing appears to have been successful, final payment should not be made until the system has been in full operation for some time (we recommend a minimum of two weeks). The stresses of volume use (*i.e.*, a large number of users) and usage spikes (*i.e.*, sudden, massive increases in use) cannot be simulated adequately in the test environment.

Also, if the system must comply with any legal requirements concerning data collection, data use, accessibility, etc., the attorney responsible for compliance must be given sufficient opportunity to test the system in the production environment. In our experience, the technical staff usually has difficulty understanding the importance and the rigidity of some of the applicable legal requirements. They may implement most of what the regulations require (as communicated by the attorney), but neglect aspects they do not consider relevant. This usually does not become apparent until close to the project's launch date when the entire system can be examined.

In sum, the final acceptance process is a critical phase of any IS project, particularly as clients tend to speed through, or neglect, interim acceptance procedures to reduce the time to launch. The terms governing final acceptance should be carefully defined and successful completion of this phase should be a prerequisite for final payment.

G. *Maintenance Issues*.

Information systems and Web sites are high-maintenance products. They require regular support and updating by experienced personnel in order to function effectively. Once a new system is in place, its performance over time will depend on the effectiveness of the maintenance services. In consequence, negotiating maintenance agreements presents some of the same challenges as contracting for IS projects, particularly:

- Developing appropriate performance standards.
- Specifying the parties' responsibilities for meeting those standards.
- Securing appropriate dedication of vendor resources (*e.g.*, resource escalation if critical problems arise and are not resolved within specific time periods).
- Using payment terms as incentives for timely performance.
- Exit strategies.

Most standard maintenance agreements are vendor-oriented and do not include tangible performance standards. However, such standards are essential both to ensure that the vendor understands what level of service is required and to support the agreement's exit strategies. It is very difficult to terminate an agreement for breach if there are no standards or the standards are vague or easily satisfied. To develop appropriate standards, first look to the specifications and acceptance standards included in the agreement governing the development and implementation of the system. If these are not helpful, follow the process described in Section III.F above for the development of project acceptance standards.

Due to the practical realities of maintenance services, often the customer's primary recourse in the event of unsatisfactory service is to change vendors. In consequence, the termination clause should permit termination for any material breach, with extremely short cure periods. Few clients can survive lengthy system downtimes.

If the original vendor of the system is the best source of support and there is no realistic alternative, penalty provisions should be considered to motivate the vendor to avoid breach in future years when the vendor may be moving on to other products but the client is still dependent on the products that it originally licensed.

Other things that should be considered in negotiating a maintenance agreement are:

- Training services, with prices agreed on in advance.
- Price ceilings for the duration of the relationship.

If the system will be locked into a maintenance relationship for some time, whether contractually or because transition to another vendor would be difficult and costly, the vendor should be required to commit to limitations on increases in its fees for at least two years.

H. *Source Code Escrow*.

Securing a source code escrow is an issue often raised in IS project negotiations to increase the user's comfort level with its commitment to a particular software product. Typically the user is provided with only the object code for the software, *i.e.*, the machine language version. Vendors are extremely protective of their source code, which is the only version of their product that is readable by humans and can be changed or updated. The source code consists of the program instructions in the original programming language before it has been translated into machine language for execution by the computer. Most software is delivered in machine-language format, which allows it to be executed directly, but makes it virtually impossible to read or modify it.

Users concerned about a vendor's long-term viability or commitment to maintaining the software usually request that the source code for the product be placed in escrow for release to the user in the event that the vendor is no longer able or willing to support the product. Typically the user pays the cost of the source code escrow, but this can be negotiated.

Even if an escrow release event occurs, however, most users will not find the source code useful, for any one of several reasons:

1. The code is not immediately available because it cannot be released until completion of a lengthy review process, thus significantly increasing the time that the user is at risk.

2. The user has no programmers on-staff familiar enough with the product or the programming language to utilize the code effectively.

3. The code is poorly documented, incomplete or not up-to-date.

4. The learning curve is so steep that it is more cost-effective and quicker to purchase a new product.

The attorney should sit down with the client and seriously assess the viability of source code escrow before deciding whether to use negotiating capital to aggressively negotiate this provision.

I. *Regulatory and Liability Issues*.

IS projects can raise a number of regulatory and liability concerns, depending on their scope and purpose. Among those most frequently in the headlines today are privacy, security and system corruption through viruses. All of them need to be addressed for any system that will be used or accessed by third parties. A brief overview of their impact on IS projects and agreements is set forth below.

1. **Privacy and Security**. Any time a system collects, handles or stores personally-identifiable information or conducts commercial transactions, users will be concerned about how that information will be handled and the level of security provided. Certain types of sensitive information, particularly medical and financial information, are subject to laws and regulations that require the protection of such information from deliberate or inadvertent disclosure to unauthorized third parties and from theft.

   If the privacy or security features of a Web site are described on the site or in any other form of advertising, the site must implement such features or risk an action by the FTC for deceptive practices. Whenever possible, broad security disclaimers should be included in the site's Terms and Conditions of Use. The functioning of the Internet depends on numerous third parties the Web site has no control over and breeds hackers who view security as a personal challenge.

2. **Data/Software Corruption**. For the same reasons as those described above, broad disclaimers of liability concerning damage to the user's information systems should be included in the user agreement for any system used or accessed by third parties and in any Web site's Terms and Conditions of Use. Viruses spread rapidly and can infect a company's system and quickly move on to users of the company's systems or Web site before either side is aware of it. They can corrupt both data and software and may force the user to shut down its entire system.

Attorneys seeking to help their clients minimize the potential legal risks of participating in cutting-edge technologies must advise them to follow the current "best practices" applicable to their businesses. This requires attorneys to stay abreast not only of new statutory and case law, but also of the potential legal implications of technical and business developments. The one certainty in this changing environment is that both the technology and "best practices" will continue to evolve.

IV. CONCLUSION

The variety of issues that must be considered in drafting agreements for information systems and Web site projects may seem daunting. In fact, they are no more complex than those we have lived with in other industries for years.

There are, however, some important differences familiar to students of historically revolutionary technologies (such as railroads or electricity). These involve the unfamiliarity of the technology, the radical changes occurring in business practices, and the relatively undeveloped state of the law. Compounding the foregoing is the rapid pace of change in all three.

The very things that make an IS development project intriguing and challenging can make the development of an effective agreement equally difficult. However, an attorney who is familiar with the business and legal issues can structure the agreement so that it facilitates the management of the project and helps the parties to solve problems quickly and productively, without resorting to litigation.

* * *

Cathy Kiselyak Austin began her legal career as a corporate lawyer, but her experience and interest in information technology quickly shifted the focus of her practice to IT-related transactions. She spent four years as in-house counsel at a Fortune 100 company, responsible for all information technology transactions. Since joining the firm nine years ago, she has focused on:

- Interactive and online content delivery, multimedia production and licensing;
- Content/software development and licensing linked to joint ventures, marketing alliances and other investment and return vehicles; and,
- Information technology implementation, enhancement, management and support.

Cathy can be contacted at (312) 569-1455 or Cathy.Austin@dbr.com.