

**FUNDAMENTALS OF NEGOTIATING THE DEAL\***  
**“DO’S AND DON’TS”**

**THE BENEFITS.** The prospect of moving from a paper-intense system of information management to a “paperless” system is daunting. If the system works, it can instantaneously manipulate data and other information (e.g. retrieve medical records, provide diagnostic templates, schedule patients, process billings, etc.) in an immediately useable format.

**THE RISKS.** However, if patient records or billing information cannot be accessed due to malfunction, it is more than a mere inconvenience; it can result in cessation of operations for as long as the malfunction continues. Worse, system malfunction could conceivably result in patients being denied proper care or even being hurt. Moreover, the sales pitch of vendors, which is often long on enthusiasm and sales fluff but short on details and commitment, can lead to a situation where the clinics are expecting more capabilities and functions than the product can actually deliver. Finally, there is a natural tension between the interests of vendors and clinics. The vendors desire to get the clinics’ money as soon as possible, drop off the software with as few commitments as possible, and limit or eliminate their obligations to remedy malfunctions. Conversely, the clinics expect the vendor to work with them to configure the software, convert data, develop custom programs, install the software, test it, train their staff, and stand behind the product to correct malfunctions and errors and provide maintenance, all for the least possible cost.

**THE MARKET CONTEXT.** The community clinics constitute a new market for software vendors. Most of the clinics have little experience in EPM let alone EMR software, the software products are not well-tested for community clinics’ needs, and vendors are marketing these products on a “one-size-fits-all” basis. Therefore, some of the ramifications of this new market context are:

- Experience has shown that software which was never particularly designed for the community clinic market has had serious limitations;
- Vendors may sense that the risks to them that the software may not perform in accordance with the clinics’ expectations and thus, are reluctant to agree to testing, or having software performance tied to specific, objective functional specifications;
- The market (i.e. the clinics’) response has been generally not to “push back” on vendor contract terms which put all of the risk of a failed implementation on the clinics. Predictably, as long as clinics continue to “give-in” to these unfair terms, the less incentive vendors have to change their ways. In other words, why should the vendors agree to take on more risk and expense if they don’t have to?

---

*\* CAVEAT: This material is provided merely as a guide in identifying some of the issues that arise in software licensing. However, because every transaction will have its own unique structure, features and issues, this form and its provisions will not be applicable to all situations and may not contain all of the provisions necessary or advisable with respect to a particular transaction. Therefore, this form Agreement is not a substitute for obtaining competent legal advice, and should not be used without review by competent counsel.*

**THE APPROACH.** As a general principle, assuming that the selected software is a “good” product to begin with (e.g. not fundamentally flawed), the more preparation invested in the front end of the process, the more likely that break-downs or unfulfilled expectations and their associated costs, can be avoided at the back-end of the process. However, this process of preparation is not easy and requires a thorough analysis of the clinic’s technology needs and expectations.

- **DON’T BE IN A HURRY!** This does not mean to be casual or not diligent. It means that you should not be such a rush that you are forced to give up essential protections.
- **NEGOTIATIONS CAN BE PROLONGED AND PAINFUL; ARE THEY NECESSARY? YES! HERE’S WHY:**

**YOU NEVER KNOW WHAT YOU CAN GET UNLESS YOU ASK.** Negotiation is a process of testing the other side’s positions. The vendor will be testing you by taking positions, and then assessing your response. For example, here’s a typical grenade that salespeople like to throw out: “this offer is only good for 3 days and to lock it in, you must sign the proposed agreement within this deadline.” If you jump at these artificial deadlines, you may have flunked the test, and the vendor will believe that you are in a hurry to do the deal without concessions. Similarly, just as the vendor is testing your resolve, you need to test the vendor’s positions by proposing your own terms which include both “mission-critical, absolute requirements and terms which you want but are willing to give up provided that you get concessions from the vendor in return.

**IF YOU ATTEMPT TO AVOID NEGOTIATIONS BY PUTTING FORTH YOUR BARE MINIMUM REQUIREMENTS, YOU MAY BE SURPRISED TO FIND THAT AT THE END OF THE PROCESS, THE VENDOR WILL HAVE CAUSED YOU TO TAKE EVEN LESS THAN WHAT YOU ORIGINALLY PROPOSED.** Why? Because the vendor will continue to test your bottom line position by pressing you. If you have nothing left to give up, and have nothing further to demand from the vendor, there is no place for you to go except (1) crashing the deal or (2) retreating from your “bottom line” requirements.

**THE ART OF NEGOTIATION IS THE EXTENT TO WHICH ONE PARTY ALTERS THE PERCEPTION OF THE OTHER WITH RESPECT TO WHAT IT WILL ACCEPT TO CLOSE THE DEAL.** Put another way, the more the vendor believes that it has to make concessions before you will sign, the greater the likelihood that the vendor will move closer to your end of the bargaining range. In this regard, your effectiveness in negotiations is dependent upon whether you have credibly communicated your negotiating positions. Therefore, as much as you might like to short-cut the process, particularly if you have an aversion to negotiating, the best results are usually achieved after a prolonged period of negotiations which (1) you are convinced that you have reached the vendor’s “bottom line,” and (2) the vendor is convinced that you will not go forward with the deal without terms meeting your “bottom line.”

- **THE IMPORTANCE OF NEGOTIATION STOPPING POINTS.** Build into your negotiations fall back positions, and always communicate credible reasons for making concessions

because if you don't, you essentially have communicated to the vendor that you have no stopping point which only encourages the vendor to take an even harder stance.

**1. ASSEMBLE YOUR TEAM.** Assemble a team to assess your technology needs to, among other things, to

- (a) Identify the functional requirements that the software must possess;
- (b) Translate those requirements into a Request for Proposal to be submitted to vendors; and
- (c) Develop from those requirements the "mission-critical" or "must have" features that you will contractually from the vendor (the "Functional Specifications").

Assemble your project team as early possible. A year or more of planning and investigation may be appropriate.

**2. CONSIDER A GROUP PURCHASE WITH OTHER ORGANIZATIONS.** If you can coordinate a group purchase among a number of non-profit organizations, it will increase your bargaining power. For example, by banding together, the deal may no longer be worth \$300,000 or \$400,000 to the vendor but over \$1 million, thus raising the stakes for the vendor.

**3. FOSTERING COMPETITION IS CRITICAL.** In general, competition yields better pricing and better terms. While you may gravitate toward one vendor or even find that one software is significantly better than other software, it is critical that you:

(a) Maintain at least one other vendor as a credible second choice for as long as possible (e.g. right up to the actual negotiation of particular deal points of the license). If negotiations with your preferred vendor go poorly or fail completely, you need to maintain options so that you are not "held hostage" by any single vendor;

(b) Maintain strict secrecy of any preferences that you have for one vendor over another. If a vendor learns that you are committed to buying its software, it knows it can take advantage of you. The process of buying a car is similar. If a dealer gets the impression that you have no intention of shopping elsewhere because you have fallen in love with one of his cars, the chances are that you will pay more than you should for the car. Never forget that every contact with the vendor's personnel is a process of gathering of information by the vendor as to what you'd be willing to accept so be careful what you say; i.e. if you give the impression that you are not knowledgeable or haven't done your homework, or that you are desperate to sign and not likely to go elsewhere, it may signal the vendor that it need not make concessions to close the deal. By the same token, never forget that every contact with the vendor's personnel is an opportunity for you to gather information from the vendor, such as after you have communicated a "no" response to a vendor position;

(c) Inform competing vendors that you are considering the products of others and that your decision will be based upon whether the terms provided by the vendor are competitive,

including the non-economic deal points such as whether the vendor provides for a detailed implementation process, acceptance testing tied to the software meeting your Functional Specifications, warranty coverage tied to the software meeting your Functional Specifications, the right to terminate the Agreement if the software does not pass acceptance testing, etc. If the vendor states that it can meet these requirements, insist that it so specify in writing in a term sheet listing the essential deal points that will be incorporated into the final agreement.

#### **4. SPECIFIC DEAL POINTS.**

(a) Demand that the software meet your “mission-critical” or “must-have” needs, and that the vendor put these in writing as “Functional Specifications.” The Functional Specifications should not only include “what” functions the software can be expected to perform, but “how” it performs such functions (e.g. response time for desired screen to come up);

(b) The Functional Specifications should be attached to, and made a part of, the Agreement because it is the baseline or “measuring stick” for determining whether the software is performing as expected or not;

(c) Make sure that the things that you expect to occur or that vendor will perform are reduced to writing and made a part of the Agreement. Bear in mind that “sales fluff,” side-emails, oral representations, and other promises that are not expressly made part of the Agreement will be excluded from the deal and will NOT be legally binding;

(d) The pricing should be spelled out in the Agreement, including what things are done for flat fees, and what tasks will be billed hourly, and if hourly, consider building in a “not-to-exceed” limit;

(e) In order to be successful, the project must be properly managed. While typical vendor form agreements provide little or no detail on what tasks the vendor will carry out in installing the software, the Agreement should specify that after signing, one of the first vendor deliverables shall be a mutually agreed upon project charter or implementation plan which divides tasks and responsibilities and sets deadlines for completion for such tasks as developing specifications for data conversion, design and configuration requirements, and custom programming specifications. The implementation plan should be attached in the form of appendices to the Agreement;

(f) Just as you would not pay a building contractor all or most of your money upfront before he has even commenced construction, you should insist upon installment payments which tie your payments to the vendor to its successful completion of milestone tasks;

(g) No technology works perfectly, and therefore, you should insist on the software passing acceptance testing, including provisions that if the software fails to pass the acceptance tests, you can terminate the agreement and get your money back (or refrain from making payment);

(h) The “Go Live” should not occur until data conversion, custom programming (including interfaces), technology infrastructure, application configuration and core group or super-user training are complete and the software has passed preliminary acceptance testing;

(i) The acceptance testing dates (and the projected date to “Go Live”) should be scheduled far enough in the future to you give ample time to prepare for it;

(j) Acceptance testing be long enough to allow for defects to emerge (e.g. one or two billing cycles 60 to 90 days);

(k) Despite vendor’s desire to disclaim all warranties, you should require that the vendor provide a warranty that the software shall conform to the Functional Specifications, requiring the vendor for a limited time (e.g. one year) to fix or replace defective software at vendor’s sole cost, or provide a refund. Just as would not purchase a car or other big ticket item without a warranty, you should not purchase complex and expensive software without a warranty;

(l) Make sure that you do not begin paying for maintenance services until the software has been accepted (passed acceptance testing), and only pay maintenance for software that is not defective;

(m) Installment payments are the norm, and don’t agree to unconditional payments— (e.g. provisions which require you to pay “on the earlier of Go Live or a fixed date,” and whether the vendor has met its obligations under the Agreement not);

## **5. DEALING WITH SALES PEOPLE.**

(a) Bear in mind that most of the compensation that vendor’s salespersons receive is commission-based. In other words, they don’t get paid unless you sign the agreement. This leads to some unscrupulous practices like promising you that the software can do everything you require without bothering to check with the vendor’s technical staff, or receiving authority from the vendor to make such a representation. If this is the case, when the contract is delivered to you, it will not include the promises that were so enthusiastically made by the sales staff. In this regard, they may be hoping that you will forget their promises, or that you will become so exhausted by the process that you will be willing to sign anything. Don’t give in to this temptation.

(b) Ask the salespeople to identify who has authority to negotiate changes in the vendor’s form agreement and ask to be placed in contact with such persons because the salespeople rarely have such authority;

(c) Don’t be bullied into submission by artificial deadlines imposed by the vendor, or “non-reasons” for vendor’s positions such as “corporate policy prevents us from changing the terms of the agreement.” For example, a vendor typically will tell you that you must accept its terms within (e.g. 5 days) or there is no deal. Don’t fall for it. Again, you can avoid this type of vendor tactics by letting the vendor know that it is in competition with others.

**6. YOUR ATTITUDE.** The negotiating process can be prolonged, frustrating and painful. In order to achieve your goals, you will have to steel yourself for it. It is important that you adopt the real threat of “walking away from the deal.” If the vendor senses that you do not have the stomach to reject an unfair offer and terminate negotiations, it will continue to maintain its hard-line position.

**7. SUMMARY: THE “DO’S AND DON’TS.”** While there are volumes of books on software licensing, here are a few “do’s and don’ts:”

***THE “DO’S”:***

- Do assemble your project team as early possible in the process of acquiring a software system to assess your technology needs, investigate the products and vendors, and interview other clinics; a year or more of planning and investigation may be appropriate;
- Do adopt a negotiating strategy which includes the real threat of “walking away from the deal;”
- Do ask the salespeople to identify who has authority to negotiate changes in the vendor’s form agreement and ask to be placed in contact with such persons because the salespeople rarely have such authority;
- Do include in your technology upgrade plan at least one alternative vendor to turn to if negotiations with your preferred vendor go poorly or “craters,” so that you are not “held hostage” by any single vendor;
- Do evaluate the vendor not only on the software’s capabilities, but also vendor’s financial stability, duration of existence, and reputation for support;
- Do build into your budget sufficient line items for retention of competent technical advisors and legal counsel;
- Do insist that the software meet your “mission-critical” or “must-have” needs, and that the vendor put these in writing as “Functional Specifications.” The Functional Specifications should not only include “what” functions the software can be expected to perform, but “how” it performs such functions (e.g. response time for desired screen to come up);
- Do insist that the Functional Specifications be attached to, and made a part of, the Agreement, thus setting the baseline or “measuring stick” for determining whether the software is performing as expected or not;
- Do insist that those things that are important or critical that you expect will occur or that vendor will perform are reduced to writing and made a part of the Agreement; “sales

fluff,” side-emails, oral representations, and other promises that are not expressly made part of the Agreement will be excluded from the deal;

- Do insist that the pricing is spelled out in the Agreement, including what things are done for flat fees, and what tasks will be billed hourly, and if hourly, build in a “not-to-exceed” limit;
- Do insist that you and the vendor mutually agree on a project charter, implementation plan which divides tasks and responsibilities and sets deadlines for completion, data conversion specifications, design and configuration specifications, and custom programming specifications, and that such documents be attached in the form of appendices to the Agreement as post-signing deliverables;
- Do tie your payments to the vendor to its successful completion of milestone tasks—think of the transaction like building a house—e.g. while you might make a down payment on signing the builder’s agreement, it would be unwise to pay him for work he hasn’t done yet or for work that is defective or which doesn’t meet the agreed-upon specifications;
- Do insist on acceptance testing, including provisions that if the software fails to pass the acceptance tests, you can terminate the agreement and get your money back (or refrain from making payment);
- Do insist that the date for “Go Live” should not occur until data conversion, custom programming (including interfaces), technology infrastructure, application configuration and core group or super-user training are complete and the software has passed preliminary acceptance testing;
- Do schedule acceptance testing (and the projected date to “Go Live”) far enough in the future to you give ample time to prepare for implementation; (i.e. you don’t want to be rushed into having to implement, and pay for, software modules that you are not ready to use);
- Do insist that acceptance testing be long enough to allow for defects to emerge (e.g. one or two billing cycles 60 to 90 days);
- Do insist that the vendor provide a limited warranty that the software shall conform to the Functional Specifications, requiring the vendor for a limited time (e.g. one year) to fix or replace defective software at vendor’s sole cost, or provide a refund;
- Do insist that maintenance services under the vendor’s software maintenance agreement commence upon the start of the acceptance testing period, but that charges shouldn’t commence until after the software has passed all acceptance testing;
- Do insist that you not pay maintenance fees for software modules that have not yet been implemented, or that are defective;

- Do insist that you not pay maintenance fees for work already paid for on a flat-fee basis or for vendor's error corrections pursuant to vendor's warranty.

***“THE DON'T'S:”***

- ✓ Don't forget that every contact with the vendor's personnel is a process of gathering of information by the vendor as to what you'd be willing to accept, so be careful what you say; i.e. if you communicate that you are not knowledgeable or haven't done your homework, or that you are desperate to sign and not likely to go elsewhere, it may signal the vendor that it need not make concessions to close the deal;
- ✓ Don't sign anything without understanding each and every provision;
- ✓ Don't be bullied into submission by artificial deadlines imposed by the vendor, or “non-reasons” for vendor's positions such as “corporate policy prevents us from changing the terms of the agreement;”
- ✓ Don't “go it alone,” without competent technical advisors and legal counsel to assist you in the process;
- ✓ Don't obediently accept vendor's form agreement which often resemble a “shrink-wrapped” “as-is” license, like “off-the-shelf” software for under \$100, without any obligation on vendor for installation, warranty, performance, etc.;
- ✓ Don't believe everything the salespeople tell you, as most are on commission-based compensation, and their focus is on closing the deal so they can get paid. In other words, be skeptical--they don't necessarily have your best interests in mind;
- ✓ Don't believe that the form of vendor's agreement protects you if the critical items that you expect to occur are not expressly in writing and contained in the Agreement;
- ✓ Don't agree to unconditional payments—(e.g. provisions which require you to pay on the earlier of Go Live or a fixed date, and whether the vendor has met its obligations under the Agreement not), and don't pay for stuff you've already paid for, like charges for correcting defective software. Before you pay, you need to be assured that you are getting what you paid for;
- ✓ Don't accept a form of agreement which doesn't specify the details of project management, like e.g. “Software installation shall be solely Customer's responsibility.” Vendor should be held to work with you (or at least be available to guide you) for each and every step of implementation, and you should require the vendor to include such services as part of the fixed software license fees;

- ✓ Don't accept a form of agreement that has no acceptance testing, warranty, or minimum standards of performance that the software has to meet to avoid being in breach of the Agreement; (e.g. don't accept that terms that the software is licensed "as-is");
- ✓ Don't accept a form of agreement which defines "Go Live" without respect to the completion of the milestones necessary for you to be ready to Go Live, such as completion of data conversion, custom programming, configuration, training, and passage of acceptance testing;
- ✓ Don't accept provisions in vendor's form agreement which are inherently unfair or which inappropriately place risks on you; (e.g. provisions which allow the vendor to unilaterally change the terms of the agreement after signing; provisions which require you to indemnify the vendor; provisions which require you to pay irrespective of whether the software functions as promised; etc.);
- ✓ Don't accept a form of software maintenance agreement which doesn't hold the vendor to any standard of timely response, or which unconditionally requires you to pay for work whether its covered under warranty or not;
- ✓ Don't accept a form of software maintenance agreement which doesn't provide for adjustment if the software is defective or has not yet been implemented;
- ✓ Don't be lax or derelict in enforcing your rights under the Agreement, such as failing to promptly notify the vendor of any defects.