

# ***Understanding the Software Contracts Process***

By John Seidl [jseidl@kurtsalmon.com](mailto:jseidl@kurtsalmon.com) and Lori Cox [lpcox@kurtsalmon.com](mailto:lpcox@kurtsalmon.com)

## **So you've decided to purchase software to improve your supply chain, now what?**

In this day and age, a company can't get by without thinking about the information systems that will make its products and services more efficient and thus, more profitable. Once a problem is identified, one of the first decisions to be made is whether to buy, build, or use pre-existing information systems resources (or some combination of all three). For several reasons, including time, money, and advances in the technologies that address today's various business issues, more and more companies are purchasing supply chain software from commercial software vendors rather than developing it themselves.

After an organization has selected a product, however, the process of purchasing the software — from the initial request for proposal (RFP), to contract negotiations, to the final software contract — can be complex, overwhelming, and exhausting. Perhaps the most frustrating aspect of the purchasing process is defining and negotiating the contract. **Understanding the software contracts process and associated risks is what truly makes an informed, knowledgeable buyer, and what results in the best solution for the least investment.** This article does not purport to offer legal advice, but rather to simply create awareness of potential issues encountered during the negotiation process.

### **The Contracts Process**

As part of the purchase process, a vendor generally submits a software license agreement as an appendix to the proposal. This agreement is based on criteria from the prospective purchaser's RFP. Many buyers assume they need to accept the vendor's standard terms and conditions as part of this process. While software contracts can be complex and confusing, there is an opportunity for buyers to negotiate to protect their best interests. These contracts are vital to users, as contracted software requires a significant investment of both capital and resources. It is very important to understand even such fundamental items as who owns the source code and whether there are any restrictions, limits, or future payments on use.

If software users are not knowledgeable about these items, or if they remove themselves from the process too early, they may find themselves in trouble later. In fact, the user's role in the contract negotiation process is critical. Users may not realize, for example, that they may make changes to the vendor agreement or prepare an addendum containing provisions that will take precedence over the standard terms and conditions. The final agreement or contract should be the result of negotiations among the users and vendor, with all attachments — including the original response to the buyer's RFP — signed and included as part of the contract. Bottom line: Treat the contract negotiation process as seriously as the selection decision. Attention, diligence, and patience will save time and money on the back end and ensure the organization receives maximum protection from risk while getting the most for its money.

## Contract Provisions from the User's Perspective

During the negotiation process, users will want to consider a number of contract provisions.

1. **Specifications and deliverables** - Articulates software specifications for functions and capabilities and precisely describes deliverables such as software components, documentation, consulting services, etc.
2. **Delivery schedule** - Details the number of days or dates for each deliverable and the vendor's and users' specific responsibilities for installation, conversion, training, and acceptance testing. The schedule may include a provision to extend vendor performance based on user-caused delays or penalties for vendor-caused delays.
3. **Payment terms** - Structures payments so the vendor receives compensation for what is actually delivered, installed, and accepted. For example, an initial payment might be 50% of the licensing fees, with another payment due upon delivery and installation, another upon testing and acceptance, and the final upon acceptance. Definitions of installation, testing, acceptance, and final acceptance are critical. Vendors will typically ask for 100% of the licensing fee and the first year's support payment at contract signing, and then request distributed payments for professional services (consulting). Consider a hold-back percentage on the professional services to maintain leverage throughout the project life cycle. Some clients even seek "pay-for-performance" terms under which the dollar value of the contract is defined based on achieved savings amounts. Obviously, this dictates that the terms and conditions of the contract be clearly articulated to define the baseline, performance metrics, and achieved gains.

An additional consideration is fixed price vs. hourly (usually labeled "time and materials") payment terms. In general, time and material contracts place the bulk of the risk with the purchaser, as the developer has no direct incentive to complete the project within a set time period unless a specific cap has been put on the number of hours allotted for the project. In contrast, the fixed price contract places the risk with the developer, as time is no longer related to the amount they get paid for the final product. Essentially, the more complex the project is in nature, the less likely a developer will work on a fixed price basis due to the difficulty of defining what needs to be built.

Finally, when defining payment terms, users should consider basing key dates on milestones vs. on the final project end date, as this will offer added incentive to complete the various project components in a timely manner.

4. **Warranties and remedies** - States who made the warranty and what may be excluded, and outlines the remedies available and procedures for obtaining them. It also includes a minimum warranty of conformance to stated specifications and documentation for an affirmed period of time. This provision specifies time allowed for the vendor to correct breaches of warranty, method of repair, location of warranty services, and any warranty extensions due to ongoing error correction. It outlines the minimum and maximum response times for warranty service, the hours of the vendor's support call center, the location of the support call center, etc.

If the buyer wants warranty support on site for start-up, this must be stipulated in the contract terms. Vendors commonly charge for software modifications, but users should be aware that most will charge again for support services related to testing and problem resolution of those same modifications. In other words, if the customization doesn't work, the buyer has to pay more for the vendor to fix it. The justification is that the issue resolution process includes a discovery period during which the vendor must determine whether the issue is the result of a software, configuration, data integrity, or user training problem. These areas are not all the vendor's responsibility and thus, their desire to be paid for their time.

5. **Testing and acceptance** - Specifies the method of testing, test data to be used, where that test data will be sourced (test data creation can be a major project), and vendor/user responsibilities. It should include a well-defined process for identifying and communicating errors. It is critical to list explicit acceptance criteria in the agreement.
6. **Support and maintenance** – Denotes the time period for support and maintenance, location of service, hours available, and limits on annual fees. Consider defining scenarios such as how remote access will be gained to the client environment if a copy of the client’s production system is maintained at the vendor’s support center, and ask the vendor to specify the skill sets of support personnel. Annual maintenance fees are typically a percentage of the original software license price – this percentage is also subject to negotiation.
7. **Documentation** - Explicitly defines electronic and printed documentation with references to versions and future documentation. This should include end user documentation (functional specifications, product training materials, etc.) and technical documentation (technical specifications, system configurations, architecture specification, interface specifications, file layouts, etc.)
8. **Software modifications** - Includes specifications for all customization work. The contract should provide for vendor or user development and user review and approval. Acceptance of the entire system, including all customization, should be a condition of final approval. Refine the rate schedule for software modification and other professional services. Billing rates for services are open to negotiation; consider a blended rate across all resources (rather than separate rates for every title in the vendor organization), and one set of rates for development work performed on shore and another set for work completed off shore.
9. **Source code escrow** - Protects the user against a vendor that is unable or unwilling to maintain the software by providing the terms under which source code and documentation will be released to the user with an independent third party acting as the escrow agent.
10. **Scope of license** - Specifies all rights, including the right to use the software for its intended purpose, keep duplicate copies, modify the source code, use software on a backup or duplicate CPU, make copies for training purposes, and assign the contract to a subsidiary or subcontractor. Special consideration should be given if there are any third-party logistics operations or other outside warehouse service providers. Vendors will typically license by number of users, by each site, or as a corporate license (unlimited users/sites). They may also implement a sliding scale – so much per user up to a certain number, then another fee level beyond that, etc. Understand that there is an important distinction between “named users” and “concurrent users” – where “named users” are individual log-in IDs and “concurrent users” are a certain number of users that would be logged into the application suite at one point in time. “Concurrent users” tends to be a better metric of actual planned activity for the application.
11. **Termination of agreement** - Gives the user and vendor prior notice (specify number of days) of alleged default, and allows the opportunity to correct the default condition. Be aware of vendor clauses to reclaim hardware or software, or to lock software.
12. **Dispute resolution** - Includes a provision to use the software while a dispute is being resolved, and specifies the location (state) of dispute resolution. The user may wish to include a mandatory, non-binding, pre-litigation mediation provision to save parties from unnecessary litigation; this allows parties to retain an arbitrator to assist with resolution.
13. **Confidentiality** - Defines confidential information. Users should attempt to negotiate provisions to protect their own confidential information (“restraint of trade” issues make vendor restrictions difficult) and examine and streamline language that protects the vendor’s confidential information where possible.

14. **Patent and copyright indemnity** - Indemnifies the user against third-party suits related to patent, copyright, trade secret, or trademark infringement. This means the vendor will protect or insure against any user liability stemming from those infringements.
15. **Risk of loss** - Passes the risk from vendor to user after software or hardware is delivered to user premises (not to carrier) and specifies damages if the loss of hardware or software affects or interrupts business.
16. **Insurance** - Defines expectations as to what types of insurance the vendor is expected to carry and requests a copy of the insurance certificates.
17. **Solicitation** - Defines restrictions around solicitations for employment of each party's employees.

It is important for prospective purchasers to remember that a vendor's product may need to be customized. Customized software also requires an agreement with user provisions, including:

- Payment mechanisms.
- Changes to scope.
- Progress reports.
- Project organization.
- Software ownership.

These provisions will protect the user's interests and create a fair and equitable agreement.

### **The Risks**

There are several risks involved with software licensing. These include non-performance, cost of modification or integration, vendor bankruptcy, and infringement. To minimize these risks, a prospective buyer may want to enlist the aid of a third party. The third party should be well versed in supply chain technology and have the business perspective and technical expertise to assist with the contract negotiation process. Once enlisted, the third party can ensure proper legal assistance is retained and help identify client issues related to contracts, such as provisions to be changed, deleted, or included, and risks inherent in the agreement.

### **Conclusion**

Software licensing contracts are complex and confusing. Prospective purchasers who are prepared and know their rights in the negotiation process can reach an agreement with provisions that protect their interests without minimizing the vendor's rights. Suitable legal counsel should review software contracts, as these purchases are very different than other types of business contracts. Ensure counsel has the appropriate experience to support the software purchasing process. Making informed, educated decisions will enable all parties to come away from the table as winners.

Written by John Seidl [jseidl@kurtsalmon.com](mailto:jseidl@kurtsalmon.com) and Lori Cox [lpcox@kurtsalmon.com](mailto:lpcox@kurtsalmon.com) of Kurt Salmon Associates in November 2007.

[Kurt Salmon Associates](#). Committed to *accelerating your results* through our *industry expertise* in global retailing, consumer products, and health care.