FINAL TRANSCRIPT

Thomson StreetEvents℠

**ACC - Basics of Software Licensing

Event Date/Time: Oct. 19. 2004 / 1:30PM ET

CORPORATE PARTICIPANTS

Karen Boudreau
Committee of the Association of Corporate Counsel - Co-Chair of IT and E-commerce Committee

Glenna Christian
Dayle & Reid & Priest - Lawyer

Diane
Committee of the Association of Corporate Counsel

PRESENTATION

Karen Boudreau - *Committee of the Association of Corporate Counsel - Co-Chair of IT and E-commerce Committee*

This is Karen Boudreau. I'm co-chair of the IT and E-commerce Committee of the Association of Corporate Counsel, and we are providing a new to in-house webcast on an introduction to software licensing. We're very lucky today to have Glenna Priest, who's a partner at Dayle & Reid -- Glenna Christian (ph), I'm sorry, who's a partner at Dayle & Reid & Priest to present with me today. She has a vast amount of technology experience, and prior to joining Dayle & Reid she was seen counsel for the U.S. Senate and Counsel of Special Investigations on the U.S. Committee of Governmental Affairs. She is a graduate of the University of North Texas and Memphis State University.

The agenda today -- I'm on slide two, is we are going to talk about the business issues, the legal issues in software licensing, product performance, product development and modification, as well as open source issues. I'm going to lead off and talk about the business issues. I will tell you that in my experience the most important thing is to understand the basics of the deal; understanding what is being licensed for what purpose and for how long is very, very important, and what the pricing is going to be.

There are a variety of different kinds of licensing; off-the-shelf licensing, licensing by function, what's the required hardware and software, will there be installation and implementation, and doesn't include open source code. It is very important, as of Glenna will explain later to understand if you are going to be licensing and the open source code, and what that means.

You need to understand the vendor; what's your leverage, what are the right flags, both as the user and for the vendor. How is this software going to be used? Is this going to be a term license or a perpetual license? What's the price? Not only for the licenses, but for the training, for installation, maintenance and support, consulting, future pricing. What if you want to upgrade later, is there a trade-in option? And how are you going to pay? Are you going to pay in U.S. dollars or in some other currency? You need to understand the warranty, and what support is required.

Let me talk briefly about Red Flag. Is this a new product? In the software world we talk about things like vaporware and spyware and PowerPoint ware. Hopefully, you want to be licensing a product that's GA, which is generally available and that is in a number of installations. You need to understand if it needs customization and modification. It's not uncommon to talk about modification needing to be many times the cost of the license. There are some software programs that regularly require five times license costs for modification. Are there a small number of installations? Is the vendor not established? Are there legal or financial issues with the vendor? Are there large dollars due upfront or a long time before installation?

Or does your client perceive there's no alternative? What's the time of the year and the time of the quarter? Software licensing is what we call in the business the hockey stick, which means many of the licenses at the end of the quarter or the end of the year. Keep in mind that that gives the user a certain amount of leverage. What's the size of the deal versus the size of the vendor? Is this a new industry your customer segment for the vendor? Do you have a previous relationship with the vendor and has that going well or has that not gone well?

The quality and experience of the vendor's team. Remember that the vendor's team negotiates together the same type of licenses and the same type of stuff agreement regularly, so you need to keep that in mind. Price versus legal terms. Keep in mind that the more difficult the legal terms, there is some likelihood that your flexibility on pricing may be less. What is your client's commitment to the transaction? If your client believes that there is no alternative, clearly your leverage is them, and also, what are the future revenue opportunities for the vendor?

Pricing. Software is licensed at a variety of different pricings. Please make sure you and your client's understand how the pricing is being handled. Is it a flat price? Is it paid up, which means that you have paid up for a perpetual license; forever, to a particular piece of software. Or is there periodic licensing? Do you pay so much per year, per month, or per term? Is it by CPU or server? Is it by user? A named user is a user that is an individual or one license, or concurrent users. That means an individual using at a particular time. Is it transaction? There are some licenses that are based on the number of transactions process by the software. Is it an enterprise license or a site license? If you get particularly a site license please make sure that you have ability to change the site or what it would cost to change the site. With an enterprise license, please make sure that it covers your whole organization. Or is it a percent of revenue? And if it's a percent of revenue, that means the revenue of your company, is there a floor or a ceiling?

Training. Make sure that you know the pricing of the training, and make sure that it -- training for particularly your technical people, and as well as the general user. Make sure you know the pricing for installation. If you need to do custom development or consulting, please make sure you understand that the pricing for the future development of customization. Maintenance is usually price on a percentage basis. It's either a percentage of the use paid, usually with an escalator, or it's a percent of the then current fee. This can be a substantial amount of money, it is not uncommon for a to be 20 %of fees paid or 20 percent of then current fee, so that means you have re-bought the software every five years. Make sure you think about pricing for future or additional products. Try to cap increases, and if you are going to be changing your data center, talk to your data center people because you may need to have two running licenses in order to do your conversion; that's called parallel use.

Glenna is going to discuss the legal issues now.

Glenna Christian - *Dayle & Reid & Priest - Lawyer*

Thanks Karen ...

Karen Boudreau - *Committee of the Association of Corporate Counsel - Co-Chair of IT and E-commerce Committee*

And I'm sorry, I'm on slide six.

Glenna Christian - *Dayle & Reid & Priest - Lawyer*

Thank you, Karen. There's a number of legal issues. Let's just first start walking through slide six, and the first one highlighted here is basically what type of license agreement is this? Is it a shrink-wrap or quick wrap or is it a bi-dial agreement where you actually have the opportunity to negotiate it back and forth? In many cases you'll find that your programmers just download software, sign a quick wrap and you have no idea what the terms are. You are not able to monitor whether they are actually using them within the scope of the license. So there's some liability issues there you want to protect and put some sort of policies in place to monitor that type of use. And a related issue is open source software licenses, a really for the most part to licenses, in that they are not separately negotiated contracts, and usually are not even quick wrap agreement; they may be just a form of a notice buried within the code, so you may not even have any idea what the terms of the license are unfilled the code is already been used. So another piece of code you particularly want to be careful with. Then in terms of figuring out really what the scope of the license is, which is the heart of the pricing issues that Karen mentioned earlier, because if you are using it outside the scope of the license in the contract, then the software vendor is going to be able to come back to you and ask for additional fees, most likely. So it's important really to understand what your current use will be and what your anticipated future use will be, and go ahead and figure that out in terms of pricing and scope now. For example, in terms of usage, it can either the limited use to some very specific types of use, or unlimited even. Usually unlimited use is going to be a higher license fee. Now, the buyer and is particularly important these days when everybody is talking about outsourcing; that's the big buzzword. Generally if you have a third party contractor who may be comes in and does some development work for you, maintains your systems for you, or if you take your systems and completely outsource the IT functions to a third party, then you need to make sure that your software licenses give you the ability to allow third parties to also use the software on your behalf. Don't assume that it will necessarily let you do that. In terms of for what purpose, you want to make sure that you cover things like testing, if you need to do product development, if you're going to do training, marketing and demos, those sorts of uses, all need to be spelled out within the scope of the use within the license. If it's not spelled out there, you can't do it, and you run the risk of infringement if you do.

Object versus binary or source is important in terms of exactly what it is that you're getting. You may think that you're getting just object code to run on the computer, whereas your engineers may think they actually need source code. They want to make sure from a technical standpoint that you're actually getting what it is that your programmers think it is that they need.

In many cases, object code licenses a sort of traditional, generally available software. It's not going to be offered on an exclusive basis. You see this more often in the context of custom develops software. If it's exclusive, one question you want to ask is if it's exclusive even as to the software vendor itself, and whether they can continue to use it or they can't even use it themselves anymore. And if so, you want to make sure that that is clear in the agreement.

And whether, if it's exclusive, that's also going to tie into things like termination rights, and some other remedies. If it's worldwide, do you have offices in multiple countries? If so, you want to make sure that you've got your territory covered in the scope. If you want to transfer it at some point to a potential purchaser, you want to make sure that covered, as well as things like how many copies can you make; only a certain number, can you make a reasonable amount? However many you need to test or backup, all those need to be specified within the scope of the use of the license.

Modification is particularly important and one that may be fairly hotly contested with the software vendor because it also ties to the issue of who owns those modifications, particularly to the extent that they are derivative works, and that's an issue that Karen will be speaking about later on.

In terms of distribution and sublicense, if it is a piece of software that you're licensing that you need to make available to your customers, or to other third parties that may be working for you, then you need to make sure that your rights of use also includes the ability to make the software available to others. I'm into the marketing evaluation and Demo as well as outsourcing uses.

Now, the general license issues on the next slide that are mentioned, you can also see some more details of these on the checklist that's available for download on the link through the web site, but for example if there's a limited territory you want to specify that. Fees, as Karen mentioned on the pricing earlier. You want to make sure every vendor understands exactly what the current pricing is, as well as pricing to cover your anticipated use in the future. Installation. Who is actually going to do the installation; will be vendor do that for you or do you have to do it to itself? If you are going to be doing it, how are they getting the software to you? Is it downloaded or other getting into you on a disk, all the sort of mechanics need to be spelled out in the agreement as well.

Software support and maintenance. Things like will they provide telephone support 24/7, or is it limited to 9 to 5? If there is a severity error, maybe its mission critical, the software goes down, you want to know how quickly they are going to respond to your call and how quickly they are going to fix the problem. And if they don't, then you may build remedies or credits or some other form of solution for failure to meet those criteria.

Indemnities, which we'll talk about later, limitation of liabilities, warranties and disclaimers. Karen will also mention in more detail later on. Term and termination. You'll want to see if this is going to be an annual license. Does it automatically renew? If so, you want to make sure you docket that, that you can send your notice to automatically renew if you have to, or know when you need to terminate it. If it is a perpetual license, and particularly that's going to trigger issues in the termination and survival. In many cases you will see a software vendor grant perpetual license, but then include the ability to terminate for a material breach.

Generally, if you have paid the premium for a perpetual license, you are not going to want to have that license terminated just for any type of material breach. A sort of a compromise that vendors and customers will usually reach is to allow for termination if the customer violates the software license terms. So in essence, if they are infringing on intellectual property the customer of the vendor, it's difficult to argue that they shouldn't have the right to terminate it. In spite of that, there shouldn't be too many other reasons to terminate a perpetual license. If so, then you may want to have some sort of refund or other remedies.

And then of course there's generally remedies in term and termination events default, if it's not a perpetual license.

The next slide is an issue as to source code specifically. Source code in many cases is also going to be tied directly to bankruptcy events. In the bankruptcy context, you are going to want to ensure that there's not an executory obligation in the contract such that through a bankruptcy, the executory obligation could be rejected. Now, an executory obligation means if you have a current rent to the object code and there is a subsequent license the source code, if there's a bankruptcy, but only if there's a bankruptcy, then the bankruptcy itself is the executory condition. Now, generally the argument is going to be that the software vendor can reject that executory obligation through the bankruptcy rights and remedies that it has as a debtor at that time in a risk jeopardizing your source code escrow or the ability to access source code at that time.

The way to deal with that, as mentioned in the slide, is to have a direct release at the time your license agreement is signed, even if you're only getting actual possession of the object code, you want to have a license to the source code itself granted at the time of the license the same time you receive the license for the object code, and that source code would only be released to you upon certain triggering events. Does maybe some sort of financial condition or bankruptcy event, but to actually have a license and of right to that source code at the time the agreement is entered into. That takes away the argument that it's an executory obligation.

Karen may want to jump in on this as well. We've sort of have the same view on terms of using escrow agents.

Karen Boudreau - *Committee of the Association of Corporate Counsel - Co-Chair of IT and E-commerce Committee*

Yes, one thing, first -- the first thing I want to say is sorry, to kind of go off track for a second, is I've gotten a number of e-mails from people who are having trouble finding the course materials, and they are in the upper right hand corner of the webcast page. It's the coming yes, upper right hand corner is this course material. Just in case anyone has any trouble finding that. Yes, escrow -- I think the most important question about source code to ask your client is assuming you got it tomorrow, could you even use it? In most cases, the answer is going to be no, so you might be fighting a very large battle over something that's not truly useful to your client.

And if they can use it, then you need to focus I think on making sure that what you get is everything your clients are going to need. You don't need just the code; you need the documentation, you need the notes, you need the tools. So if you are really serious about wanting source code, then really go off. Can consider getting the source to a third party or to your people just long enough to make sure it can be compiled to create the object code that you need. It's going to be an expensive and time-consuming process, so just know that. That's sort of all I had to say.

Glenna Christian - *Dayle & Reid & Priest - Lawyer*

Well, I completely agree; many of the software programs now that are being licensed are really cobbled together pieces of third party software, with maybe some proprietary code weaving it all together, and the only piece that may actually go into an escrow would be that limited amount of proprietary code called in together on the third party software, and without that third party software, you wouldn't really be able to run the program and so your client could end up having to go out and repurchase already license all that third party software. Additional time, cost, and expense. So as Karen mentioned, it may not even be, from a practical sense, worthwhile to do it, but it's something that's usually hotly contested.

One last point on that, the triggering event, bankruptcy issues are always slightly iffy, and I never want to rely on them. I would rather have a triggering event that is going to occur before a software vendor files for bankruptcy, and you can usually start to see them having trouble manifest themselves in other ways. For example, they stop providing technical support on time, they are not providing software maintenance like they should be. So if you have a good detailed description and a schedule of requirements of what has to be provided for technical support and software maintenance, you can tie the triggering event in your source code escrow provision to those software maintenance and support, and be able to trigger the release prior to them actually going into bankruptcy.

On the next slide is just some little points with respect to confidentiality provisions. Usually, these should be mutual. The general standard for maintaining confidence and so be maintained in the strictest confidence in accordance with your own standard practice. Generally, you allowed third party disclosure if there's a written agreement. One thing that you may want to add that you don't always see in confidentiality provisions, is that you can also allow disclosure to outside consultants, if they enter into a written agreement or bound by another confidentiality obligations. For example, if your outside counsel needs to review the license permits for some reason, you generally not going to ask your outside counsel to signed a separate NDA just to look at the software license. They are already ethically bound to obligations that are even more strict to this. So you add that language in there, then you just -- one last piece to have to worry about. If you setting up the data room for due diligence or something along those lines.

Timeframe on notice, nondisclosure and trade secrets I also want to specifically mention as well. You will frequently see anywhere from 2 to 10 years on an obligation to maintain the confidentiality of whatever confidential information is. One piece that you want to carve out, particularly if you have some information that you'll -- and trade secrets that you'll be disclosing is that to the extent you disclose trade secrets, you want to carve that out from the time period because trade secrets are protectable indefinitely under state trade secret law, and if you put a limited period 2 to 10 years on the obligation to maintained that confidentiality, if you disclose trade secrets within that they are no longer are obligated necessarily to maintained that confidentiality, and you could lose your trade secrets as well. So carve out of that time period the trade secrets that may be disclosed as well, and include an obligation to maintain that as long as possible into absolute trade secret law.

Generally, you want the confidential information to have to be marked. You don't want to have to guess as to whether it should be treated as confidential. You will frequently see oral or non tangible information. Sometimes you'll see them also added that's reduced to writing, and subsequently marked as such. I'm even reluctant a lot of times to put that in there, depending on who was providing the most information because she you can end up in just a battle of he said she said, so if possible I tried to avoid including oral information. And then there should be the usual exceptions. For example, information that's in the public domain or that provided with no breach of this agreement.

As to other legal issues that you'll see in the checklist that's in the materials, there may be an audit, particularly in terms of payment if there is any revenue based payments. You may receive prohibitions against publicity or the software vendor may want to list you as a customer if they want to use your logo or any sort of branding on their marketing materials and you probably need to include a trademark license and you may want to have approval rights over what they're doing, particularly so you don't want to give them a naked trademark license, but they want to exercise particular control over what they're doing with their logos.

On governing law and venue, you know, every software vendor is going to want to have governing law wherever they are based. Let me just had one point on that. There was a fairly recent decision from New York that doesn't address conflicts of law principles, which is you'd be typically carve out of governing law, that addresses statute of limitations and treats it just like conflicts of laws principles. So even though the contract provided for New York law, the statute of limitations ended up I think applying Delaware law or a law of a different state. So it's like a conflict of law issue with a ride, so that may be something else that you want to start carving out of those clauses now as well.

Insurance. If they are going to be on your site providing any type of training or support, you may want to make sure that they've got standard insurance, aside from that, I just may want to mention that if you know for example that your company may be looking at an exit strategy, with an asset purchase, then you want to carve out the ability to assign to a potential purchaser so you don't have to go back and have approval and consent on the transfer law of the software licenses.

Karen, I want to make sure we don't run short of time, so I'll turn it over to you for product performance.

Karen Boudreau - *Committee of the Association of Corporate Counsel - Co-Chair of IT and E-commerce Committee*

Yes, one thing on product performance and acceptance is I want to make the point on the software revenue recognition rule. If you're doing a lot of software work, a good use of a few hours of your time is to read the software revenue recognition rules. Because it is important to software vendors to be able to recognize their revenue. So if you commit to a license for $1 million before the end of the fiscal year, that they can recognize that revenue in their earnings.

I can get anybody who has the link to the revenue recognition rules. What I usually do is I asked my accounting people to get it for me because they do change pretty regularly, but it is important to understand the revenue recognition rules because it is not uncommon in negotiations for a vendor's lawyer to say, "I can't agree to that because it's a violation of revenue recognition rules." Very occasionally, that has not been exactly correct in my understanding of revenue recognition rules. We could have a debate about that, but if it isn't really a revenue recognition issue, then I just becomes a matter of negotiation.

In acceptance, you've received the software, it's then installed, and you need to accept it. Many of the licenses will say you accepted within a certain period of time. For instance, when you begin productive use or after -- a period of time after acceptance, or the most difficult is operates within specifications for a period of time. That gives you an opportunity to test. Testing can either be unit testing, which is testing by module or by system or by facility. It can be system testing, which means it works in the whole system or it can be component testing.

I'm now on slide 12. Under warranty, warranties vary very much in the software industry. They can be 90 days or one year. You will not normally receive the warranty that is for the life of the time you have the product. And there are revenue recognition issues for that, the reason for that. What you can get is a requirement under your support services sometimes that issues will be fixed within a period of time, but any severity ones which are most of your issues will be fixed within a certain period of time, or you'll be provided with a work around.

One of the least beneficial to the customer warranties is a warranty that says only that the media will function. So if you get the CD, the CD will not have a defect, and all you can get is another CD irrespective of whether or not the software works. What you'd like to see is that it will meet the published specifications without a substantial problem for a period of time and that it will be fixed within a period of time that the most serious issues, they are usually priority 1 or priority 2 issues, are fixed with severity 1 and severity 2 issues are fixed within a very specific period of time. There can be other warranties; warranty of title, warranty of compatibility. That means that this software is compatible with another software, that it works in this configuration. You are going to configure it to a particular kind of hardware, and it is also not uncommon you should understand for there to be -- for you to load the software and discover that you need a substantial amount of new hardware. You need to cover that and make sure your clients understand that, and that it is capable of performing in a certain way.

There's an often a compliance with laws warranty. You will probably not receive a most favored customer warranty; that is, that your client will receive the best pricing terms and conditions. You can try to get it and maybe you will. It has to be a substantial agreement, and you may have to trade some things that might be more important to your client. There are certain year 2000 warranties. There are warranties that the documentation will be accurate and complete, that the software will respond in a particular time, that it will have a certain capacity that it can process a certain number of transactions.

One that you should probably make sure that you get is a warranty, or at least a representation and a requirement to fix if there are bugs, viruses, time bombs, or some kind of what they call elicit code in the software. If there is an automatic shutoff in the software controlled by the vendor, you should make sure that you understand what it is and your clients understand when it can be used. And then what is the remedy for warranty.

Support. Support will virtually always be provided, and as I said before, it's quite expensive. It can be not uncommonly 20% of the cost of the software. You need to understand what's the scope of the support, want kind of -- how will they provided it and when will it be provided. Is it 24/7 worldwide? Is it 9 to 5 India time? You need to understand is it only e-mail, and if that's -- you know, whatever it is, if it's OK with your clients and you're sure they understand, that can be OK, depending on the criticality of the software.

Make sure you see whether it's response time or repair time. Response time is how soon they start working on the problem. Repair time is when it is fixed. So most of the time you'll see response time, but as I said before, you should try to get a repair requirement on at least the two most severe kinds of problems; severity 1 and severity 2 issues. Severity 3 and severity 4 issues; severity 3 it's not a substantial impairment and there is a good work around available. Severity 4, it's virtually all impairment. It'll maybe be addressed in the next release.

You need to understand if you're going to get updates, upgrades, new versions, or new releases. And if you're not getting those, that could be important. You should also understand if you are getting You versions and new releases and you've got and customization are you going to need to re-customize the software to deal with the new versions and new releases. And again, you need to know what your remedy is going to be for failure to provide support.

The warranty disclaimers. Don't be surprised to see a disclaimer that there may be notice in the software you won't have error corroboration. A disclaimer for warranty or merchantability, a disclaimer for fitness for a purpose, sometimes a disclaimer for title, and sometimes a disclaimer for non infringement. If you get a disclaimer for non infringement you probably won't get it out, but please make sure you get a very good indemnity to cover that.

The triggers for remedies; what's the notice requirements? Is there an opportunity to cure? What is the escalation procedure? And the remedies can be repair, replace, or refund, particularly watch for refund. If it's depreciated over 3 years, then there is effectively no remedy at the end of the third year. And what you get back purchase price and sometimes you will see liquidated damages. Which can be relatively rare, but you can sometimes get amount of money paid under the agreement that you actually used to replace the software with something else. For services you most commonly see re-performance of a service, sometimes, again, use the customization.

I'm on slide 15, and Glenna is going to talk to you about indemnification.

Glenna Christian - *Dayle & Reid & Priest - Lawyer*

As Karen mentioned, indemnification is particularly important if there is a disclaimer of warranty for infringement. In many cases you'll see this being mutual, and really question whether you, as the customer of the software need to be providing indemnification, take a look at your use, take a look at the price of the software and see whether it's something that you really want to get or have the leverage to exclude in this case.

Some of those things that you want to cover within your scope of indemnification action or the in action of the software vendor, their suppliers or employees, their contractors, this would cover in connection with the performance of the services or just generally anything else they are supposed to do or should not do. You want to include any claims of their employees or subcontractors. Personal injury, property damage, if there is any misappropriation of your confidential information or breach of the license grant.

One thing to note in terms of indemnification for breach of the agreement, including breach of a license grant, generally your licenses, your agreements, are going to exclude a liability for indirect damages. If you have indemnification that covers breach of the agreement, any provision of the agreement, the court may award damages that generally would have been excluded as indirect damages, but because you have it covered within your indemnification, you may still be obligated to pay that. It also works the other way around, if the vendor is the one that's obligated for providing indemnification, but just bear in mind that the remedy that a Court may order might include indirect damages, but for purposes for indemnification clause, actually be deemed direct damages. So that's something you'd want to be careful for.

And of course obviously one of the most important is going to the intellectual property rights, there's no infringement. Now, generally you want to cover patents, copyright, trademarks, trade secrets, or any other sort of analytical property proprietary right. What's going to be particularly important for you is the scope and the territories mentioned here. If you are using the software in multiple locations and multiple countries, your best protection will be to make sure that the territory for which indemnification is covered is through every country in which you are using the software. And generally your software vendor will tell you, "We haven't done patent searches in this country, so we are not going to cover you for patents, we will only cover you for patents in the U.S.". And that may or may not be a business risk that you're willing to take.

Copyright they may try to limit as well. Generally, most software vendors though will agree to give you indemnification to the extent that it's a country that's a party to the Burn convention. And then maybe other limitations on that as well. If you are the software vendor or as you as the customer getting indemnification of non infringement, you also want to make sure that it is appropriately narrowed and that you're not giving a broad indemnification for any infringement in any country.

On the next slide, the remedy and exception to the indemnification or infringement of the software, generally you will see some sort of obligation for the vendor to obtain a license for the infringing piece of the software to either repair or replace it so that it is non infringing, and if they choose that route you want to make sure that whatever that replacement is or repair provides the same or substantially similar functionality. It's not going to do you a whole lot of good if they fix it, if the software no longer performs the way that you had anticipated or in accordance with the documentation. And as a last resort either terminate the license and refund. And as Karen mentioned earlier, there are issues in terms of the refund, with depreciation, and what you may actually receive.

In some cases, the license, you will see that these remedies are the sole and exclusive remedy for infringement of the software. As the customer, you are not going to want this to be fully inclusive remedy, you are also going to want the indemnification as well. It should not be your only option. Some conditions to these remedies is that there is timely notice of a claim of infringement, that the software vendor has full control over the dissent and that you as the customer reasonably cooperates with them and that should be at their expense and not yours.

And there are generally some exceptions to their obligation to indemnify. If you as a customer are using the software in a way that makes it infringing or you modify the software in a way that it then becomes infringing, those are pretty standard exclusions from the vendors obligation to indemnify you. There maybe some issues in terms of, well, the modification was done by them, or they approved it or the third party that they had recommended.

So you want to make sure that you've got the carve out from the carve out almost and be very specific as to those exceptions to the obligation for indemnification. There may be cases in which it actually is more appropriate for them because they are controlling the modification or the use of the software.

The next slide on limitation of liability. You're going to want these provisions to be reciprocal as well. In many cases, you'll see the license agreement that you receive, that it's just one way and only protects the vendor. Many companies now have policies that do not allow for contracts with unlimited liabilities, or that there's a maximum amount, the potential liability end of the contract.

So you want to make sure that your limitations of liability cover that. So consequential damage as I mentioned before, the limitation of liability for indirect consequential, (inaudible) like that, is something that's going to be fairly standard. However, you want to make sure that you carve out from that the indemnification obligations if there's a breach of confidentiality or the license agreement.

In many cases, the damages for those types of breaches would really be indirect damages and you want to make sure that the vendor is still going to have to be responsible for paying those. And the same is true with respect to the direct damages. There are many different ways that you can come up with a cap on the aggregate damages under a contract.

In many cases, you will see it based generally just on the fees paid under the agreement, whether that's the fees paid within the last year prior to the event that gave rise to the liability, or all the fees that have been paid to that point under the contract. But you'll generally see some formula based on the fees under the agreement for the cap on damages.

In some cases, it's tied to the limits of insurance, or you may see just a fixed amount, $5 million, $10 million, depending on the value of the software license.

Now I'm going to turn it back over to Karen for the issues in developing and modifying the software.

Karen Boudreau - *Committee of the Association of Corporate Counsel - Co-Chair of IT and E-commerce Committee*

Great. Many times in situations, you're going to be faced with there being some kind of custom modification necessary to the software that you've licensed. In my sort of 20 years of experience in doing this, this is the situation which has way more often than any other led to a disagreement between the parties.

And in most cases, simply because the vendor in all good faith believed that the task was one thing, and the customer in all good faith believed the task was another. The customer usually in particular was in a very big hurry to start without defining the goals and objectives clearly enough and the responsibilities.

If you spend time creating a detailed and accurate statement of work at the beginning of these projects, I found that it saves you a lot of time and money at the end, and for both sides. Vendors no more want to have a dispute with a customer than a customer wants to have a dispute with a vendor.

Make sure that you face the development task from the deliverables and have this all in writing so that both parties can understand what it means. A timetable, what's going to be done, other milestones. Are there payments based on particular milestones? Or are there consequences for failing to meet? I've probably been most successful in very large development projects.

We used (inaudible), not fixed. We gave a percentage additional to the vendor if they finished early, different percentages by different amounts of time early. How would you accept the deliverables? Please make sure that your clients are prepared to take the time to analyze the interim deliverables.

How a project changes the handle. This should be dealt with in the statement of work, both parties should understand. You need to identify project problems. There will be project problems. How will they be identified?

I sometimes use a clause that requires both parties, they're usually regular meetings, to present a problem within a certain number of meetings after it has been discovered because if problems that are allowed to linger, they don't age well and they get continually worse. You can sometimes get what's called an independent obligation to continue performance.

I definitely call that my, you can't take your bat and ball and go home. If this is a substantial development project or a substantial customization project, sometimes you can get the vendor to agree to a transition phase to another vendor even if they or you decide to terminate. They will often require payments up front however, so be prepared.

And what's the escalation cap and procedure if there is an issue? Who in your organization and the vendor's organization will be available to address problems? Personnel issues are very, very important. You may want to get a skill set description. You want to make sure that they don't sell you on the project with the A team, reassign the A team and you get the C team.

What's the assignment? Are there firewalls if they're doing developments for one of your competitors? Can they be assigned to your competitor? What are the key personnel that you'll say shouldn't be assigned during a project, reassigned during the project? Can you as a customer replace for cause or without cause?

If you have difficulty with a particular consultant, can you have them replaced? What is their training? Of course, they need to comply with your facility rules. Vendors will often ask you and you may also want from them, a non-solicitation clause. You might want to consider direct versus non-direct solicitation. Indirect solicitation would be, you put an ad on Monster and they responded.

Particularly with a large company, I've been at large retail companies, how could we know that somebody who might work for a consulting company didn't get a part-time job in some store over the holidays? A time restriction, or if the end of the agreement (inaudible) time or as a period of time after they leave the assignment.

You probably want the non-solicitation to include your independent contractors and the vendor will want to include theirs. The remedy can be liquidated damages, a percentage of the salary or a direct prohibition. Development. One of the major issues in development is who is going to own the intellectual property rights?

If you want to own your intellectual property rights, please make sure that you get a work for hire clause and an assignment. I usually like to also get a license for anything that can't be assigned and depending on where you're doing business and with whom you're doing business, you might also need a moral rights clause.

Don't be surprised if the vendor wants the ability to independently develop or a residual rights clause. You may have no choice but to accept that, but make sure that it works for you. Don't forget to get a license to the pre-existing materials which will be included in your software.

You are unlikely to get an entirely new development, so make sure that you have a license to any pre-existing materials that are included in the numbers provided to you. Make sure that you have a further assistance clause, meaning that you've taken the actual property rights and you've entered complete documentation to help you with filing.

Pricing. These can be priced on a fixed price into evenings, not to exceed work time and materials. What will work best depends on your particular engagement, but please make sure you know what you're getting. The rates can be hourly, daily, or blended rate.

That means a rate for a certain amount depending on the stair level of a (inaudible), $300 for any level of employee. Find out if you need to pay expenses, telephone, overhead, what are the travel and expense expenses? Decide what the payment schedule is going to be.

Are you going to pay them a certain amount monthly? They'll send you a bill, you pay them? A certain amount in advance? It is not uncommon to have a substantial advance payment. Obviously if you are the customer, you'd like that number to be as small as possible. Will you hold back? Maybe you pay monthly, but it's 70 or 80% or 90% of the fees incurred.

Or will you only pay upon delivery or acceptance? I provided a general outline of a statement of work, description of the project, who is going to manage the project, the roles and responsibilities of each party. When you look at these, look for the catchall phrase about who's responsible for everything else and see, but make sure that if your client is responsible for anything not previously discussed, that they are prepared to take that on.

What are the assumptions? There will be likely a long list of assumptions provided by the vendor, that is, these things must be true if we're going to provide this development over this period of time with this fee. Spend your time reading those assumptions very, very carefully and be adequately certain that your client can comply with them.

What's the pricing, incentives, what the daily damages or fallback are all possibilities. What's the staffing? We talked about the different choices of that. Project plan and timetable. What are the documentations and programming standards? What is going to come out of the other end?

Are you going to get fully documented code or are you just going to get some object code that fits into your existing object code? What are the critical milestones?

And (inaudible) will talk to you about the very complicated matter of open source issues. I'm on slide 22.

Unidentified Company Representative

Now open source issues now could almost spend an entire presentation on their own, but I'll try to give you the highlights here in about 5 minutes or so.

Open source has particularly come in on everybody's radar screen in the wake of all of the SCO litigation against IBM regarding the Linux software, but not necessarily a vendor understands what open source is or what it means and what the issues really are. Open source software is generally the right to use source code to a program.

The difference from all of the different software that we've been talking about so far today is that it doesn't have those restriction protections on the software from the vendor in order to maintain the proprietary nature of it. The thought is generally that open source, everybody should be able to use it, modify it, and do with it as they wish.

The one feature that is a little unusual depending on the type of open source license is, what happens if that open source is distributed? Now the license that most people think of is the GNU general public license and it's sister the LGPL, the lesser general public license.

And generally under the GPL, what happens on a distribution is that the customer also has to distribute the source code to the program to all of the third parties. The trick becomes, however, if you have taken that source code and somehow created a derivative work and that is the work that's distributed, the derivative work may essentially be a de-blinking of the open source software to your own proprietary code if you've developed something on your own.

And then they would be available to your customers depending on how the software is linked, that may be sufficient to trigger these distribution requirements and require you to distribute even your own proprietary software in those source code forms under the same terms of the GPL.

Now keep in mind that only applied if you're distributing the software and there are questions offered on what distribution means. There's a fairly recent case, it's again also in New York, called getaped.com in which the court found that just simply right-clicking on a Web site and be able to view source, and the context of that case was actually a distribution.

So if for some reason you're using open source software that you've licensed and pieces that code may be available through the Web site and viewing the source feature, that in and of itself may be a distribution sufficient to trigger this requirement. So don't think of distribution in a normal context of just handing over a copy to your customer.

If you have an affiliate, if you have third-party vendors and you give them a copy to use in connection with some development work they're doing for you, that also may be a distribution that would trigger these requirements. Now the GPL's I mentioned, depending on whether you've created a derivative work, obligate you to distribute the source code to your own programs to anybody that asks for it, not just your customers.

There are other open source licenses out there that don't require that. The LGPL is one of those in which you're allowed to use libraries without having to do these similar obligations. Apache is another one that's very common and there are just a number of them out there. So don't assume just because you hear open source that you've got these obligations on the distribution.

You want to actually take a look at the particulars, the form of open source software license that's being used and how you're actually going to be using them in your organization to determine what sort of issues there may be. As you see on the presentation, there are a few recommendations in there to help sort of minimize the risk generally.

The other feature of open source software, unlike proprietary software that you license, disclaims all warranties and limits all liabilities on the part of the vendor. They are making it available to you to do whatever you want to, as long as they're not going to be responsible for it.

Now, if you're using it in the context of a mission critical application, this may not be something that you want to do. You have really no third party to take recourse from. If this offer is infringing or it fails, you're not going to get technical support and maintenance for using it.

The other piece that you want to keep in mind is if you are distributing it to one of your customers, for example, as part of a bundled package of services and software, you want to make sure whatever warranties and indemnification you give to your customers don't include warranties and indemnification for the open source software.

In the same context, when you're licensing from a software vendor, one of the things you want to find out from them is what third party software they are using to the extent that it includes open source software. That's something you need to know and just be able to assess the risk depending on how you intend to use that software.

A good way of flushing that out, because many times they won't tell you, is to in your revisions back from in your software license, include a breach of warranty that there is no third-party software. That will at least usually start the discussions on what sort of software is being used within their program.

The consensus you can take is if you're going to be doing large scale development work in houses, they want to form an open source software committee then add some of your technical and legal representation, (inaudible) the due diligence of software before they use it, require approval before any open source is used.

As I mentioned, you're not going to have to necessarily go through a click wrap or enter into a bilateral agreement in order to have the open source license. It's simply a notice in the text file. The other piece to keep in mind is if it's purely a license as opposed to an agreement, then you have issues with it just being a license. Can it be modified unilaterally?

Can the software vendor terminate it unilaterally? If it's not a contract agreed to between two parties where no modifications or changes can't be made without the other party's consent, then there's a little bit of a weakness that you may not want to rely on depending on how you're using it. You also may want to limit the number of off site people who can use open source.

You want to make sure that people have training and who are understanding with these issues are the only ones using it, influence and guidelines in the policies on what they can and can't do with it, to make sure you don't trigger a distribution unless you have approved it and taken steps to protect against that. And then you want to have some internal mechanisms to monitor and control.

There's some software you could license out there now that will run tests through your systems to determine whether there is any open source on there that you didn't know about, and anticipate future transactions.

If you're offering software as part of your packages, and at some point down the road you may be looking at doing an asset sale or firing somebody, then you're going to want to make sure that you've documented your open source software use, so that you can be able to show a potential purchase or you as a potential buyer can understand whether there has been any potential disclosure of proprietary code unintentionally.

And that's it for open source in less than 10 minutes.

Unidentified Company Representative

And we have one question that's Glenna is going to address. If there is no express provision in a license that covers the agreement against the customer's name and logos and to locate the list of names in a customer list or use the name and logo on a Web site, whether or not that that's trademark or a trade name?

And we have some other questions, but I'm going to have Glenna ddress this one first.

Glenna Christian - *Dayle & Reid & Priest - Lawyer*

Sure. If you're using a third party's logo or trademark without their permission, then you're going to run into issues of trademark infringement. If you're purely using the name, then the context of writing a description about what you do and you're not using it in a marketing context, then you may simply be referring to them as a trade name and not using their actual trademark.

It's not an issue if you're using a trade name in a non-marketing context. If you're using their logo, however, on a Web site, then you've got an issue and you need to get a trademark license for that.

Diane, do you have any other thoughts on that?

Diane - *Committee of the Association of Corporate Counsel*

No, I absolutely agree. We had one comment that I wanted to point out. When you're looking at an escrow clause in the release or some release of the source code, don't forget to get the non-solicitation provisions also removed so that you can hire one of this company's software engineers if you need one.

And also, you need the ability to disclose the source code to your independent contractors, of course provided that they would keep it also confidential. I'm sorry?

Glenna Christian - *Dayle & Reid & Priest - Lawyer*

That's a very good point.

Diane - *Committee of the Association of Corporate Counsel*

Yes, I thought it was great. There's another question, how do you deal with situations in which your vendor says it won't provide maintenance unless licensing installs new releases? I concurrently understand the vendor's point that they can't support releases forever.

I often have a clause that requires that there will be no substantial modification of the software necessary in order to implement the new releases, and that if there is, they will continue to support the previous release and then I put in a substantial time period for my client. Or alternatively, I will allow the vendor to do the modifications for free.

This happens if you have highly modified software and then you try to add in the new release if it's been highly modified, it could be substantial further development work.

And this is the last question. What if the vendor refuses creditor refunds for failure to respond a cure for a problem with the software under maintenance support? At a minimum, if I absolutely have to do business with this vendor, I would make it a minimum of a breach of contract and make sure I have a substantial limitation of liability.

If they're really taking our position and they won't step up to somehow making - at a minimum responding, preferably making a fix, I would really want to do a substantial background check on that vendor as to whether or not they've had any issues or substantial issues and look and understand what my alternatives to that vendor might be. That would be my defense.

Glenna, do you have anything to add on that?

Glenna Christian - *Dayle & Reid & Priest - Lawyer*

No, I totally agree. I think that's your best protection.

Diane - *Committee of the Association of Corporate Counsel*

In summary, I really hope that this has been helpful to everyone. I appreciate everybody's time. You can continue to e-mail me with any questions that you might have and I'll draft Glenna to help me too. She just found that out.

Glenna Christian - *Dayle & Reid & Priest - Lawyer*

Certainly. I'm happy to do that.

Diane - *Committee of the Association of Corporate Counsel*

And you'll note that this Web Cast will be on the Web site for a year. We'd like to invite anybody who is coming to the annual meeting to participate in the Information Technology and E-Commerce Committee Meeting, get up bright and early on Tuesday morning at 7:30.

We'd love to see you there, and of course, we'd love to have any of you that would like to participate in the IT&E Commerce Committee.

Again, thank you very much and this is the end of the Web Cast.

DISCLAIMER

Thomson Financial reserves the right to make changes to documents, content, or other information on this web site without obligation to notify any person of such changes.

In the conference calls upon which Event Transcripts are based, companies may make projections or other forward-looking statements regarding a variety of items. Such forward-looking statements are based upon current expectations and involve risks and uncertainties. Actual results may differ materially from those stated in any forward-looking statement based on a number of important factors and risks, which are more specifically identified in the companies' most recent SEC filings. Although the companies may indicate and believe that the assumptions underlying the forward-looking statements are reasonable, any of the assumptions could prove inaccurate or incorrect and, therefore, there can be no assurance that the results contemplated in the forward-looking statements will be realized.

THE INFORMATION CONTAINED IN EVENT TRANSCRIPTS IS A TEXTUAL REPRESENTATION OF THE APPLICABLE COMPANY'S CONFERENCE CALL AND WHILE EFFORTS ARE MADE TO PROVIDE AN ACCURATE TRANSCRIPTION, THERE MAY BE MATERIAL ERRORS, OMISSIONS, OR INACCURACIES IN THE REPORTING OF THE SUBSTANCE OF THE CONFERENCE CALLS. IN NO WAY DOES THOMSON FINANCIAL OR THE APPLICABLE COMPANY ASSUME ANY RESPONSIBILITY FOR ANY INVESTMENT OR OTHER DECISIONS MADE BASED UPON THE INFORMATION PROVIDED ON THIS WEB SITE OR IN ANY EVENT TRANSCRIPT. USERS ARE ADVISED TO REVIEW THE APPLICABLE COMPANY'S CONFERENCE CALL ITSELF AND THE APPLICABLE COMPANY'S SEC FILINGS BEFORE MAKING ANY INVESTMENT OR OTHER DECISIONS.