A How-To Guide for Negotiating Tech Deals

I started my career in a mid-sized law firm doing sophisticated corporate deals about 28 years ago. Over 20 years ago, my practice evolved into doing sophisticated tech, telecom and outsourcing deals. With this evolution, I found new contracting norms. The most significant was that I went from a world of good first drafts prepared by competent lawyers to a world where first drafts often came from illiterates. This article is a how-to guide to get you from illiteracy to a workable and fair agreement.

Unfortunately, sales teams dominate my tech deal world and they think that a good contract is the last one they did with the names changed. After all, that’s the path to a commission check this quarter.

I would comment that most first drafts from tech vendors (including brand-name vendors) are atrociously written. They arrive on thoughtlessly used templates modified by the incompetent. They are not so much one-sided in favor of the vendor as simply not an expression of the deal.

The norm in my world is that in my first pass on a proposed agreement, I will have 5 to 15 comments and redlines per page. Now, consider a hypothetical 30-page agreement (without attachments), and simple arithmetic tells you that we have 150 to 450 issues to discuss with the vendor.

Some of the discussion is substantive like discussing intellectual property indemnification or carve-outs to the limitation of liability. Much of it is about defining the core business terms like what the buyer is buying and what it will cost. (If you think tech agreements could not possibly leave out core terms like this, you don’t live in my tech deal world.) Then, a lot of the discussion is about rewriting our way through the illiteracy of the first draft. In this category, many of my comments consist of, “What does this mean?”

What are the Goals?

I think that it is important that we define the goals for the negotiation of our hypothetical deal. For most deals, I define the goals this way.

Let’s imagine that everyone sitting at the negotiating table is not available to interpret the agreement when a question arises. The written document has to be good enough to stand on its own. It must explain the deal sufficiently such that a substitute team with the same training as the original negotiating team can understand the intention of the parties.

The underlying point, beyond the obvious goal of avoiding the need for parol evidence to foster interpretation, is that tech contracts don’t need to define every complex tech concept in such a way that a hypothetical high school educated juror could interpret
it. Rather, the assumption is that interpretation would be by a team capable of understanding a technical contract and could dumb down the explanation as needed.

Another goal is simply to use the contracting process to insure communication between the parties. I still find it amazing when I start probing a description of services only to discover that the parties have so few overlapping ideas on what the contract is about. Then, when they do agree, I find myself often saying: “That’s great and we agree, but the contract does not say that. Can we reword it to simply say when you just said?” (All the while, I can feel the sales team seething at me because of my absurd requirement that the contract accurately state the deal.)

Lastly, I think another goal should be to pull those one sided vendor terms back to the middle. Thus, you need to focus on things like limitations of liability, indemnity, choice of law, acceptance testing procedure, and so on.

Norms in the Industry

The tech world has some rather firmly established deal norms. It’s important to know those so you don’t push where pushing is not likely to garner meaningful concessions. Knowing the norms allows you to effectively focus on the areas where you can have impact while choreographing your concessions around areas where you’re not likely to win the battle anyway. You simply cannot negotiate these deals effectively unless you understand the norms.

For example, the IT world has developed some unusual norms with warranties and allocating risk. In other industries, these might seem absurd, but not in IT world.

As a preliminary matter, let's clarify that I'm using the broad term "IT" in a broad sense. The same principles will apply whether we're discussing outsourcing your entire IT department, hiring somebody to design a website for your company, buying hardware, or just having some software customized for your company. Whatever it is, many of the same contracting norms apply.

Warranties

If your new car doesn't work, you take it to the dealer for repair. Knowing what it's supposed to do is easy. After all, it's a car. Everybody knows what a car's supposed to do.

It's rarely that easy with IT-related contracts. Certainly, it could be that the computer doesn't turn on or that all network communication is down. These would be easy cases. As long as you have anything that looks or smells like a warranty in your agreement, you should be covered.

The problem arises because it's just never that simple. Typical IT problems are more like "the computers are too slow," "the network is too slow," "the system crashes
too often," or "the website doesn't have some of the functionality we expected." Moreover, these are the types of problems that can lead to ugly disputes.

There's no magic after the problem arises. The solution is careful contracting at the front-end. For example, if you don't take the time to quantify the speed you expect from your network, where do you go with “it’s too slow” when the other side says “no, it is not too slow.” If you lack an objective standard, you may just have a loser on your hands.

While, there is no doubt that taking the time to carefully contract for IT services slows the date of the contract's signature, the only other choice would seem to be a wing and a prayer. Contracting, like any other process, simply takes time. Still, it can save your company if things go less than perfectly.

Using the custom development of a complex software product as an example, the norm in the IT industry is a vaguely worded warranty that says that the software will function according to its specifications, or some other vague document or attachment to your contract. The point is that the standard the software must meet is this "other document."

It's the norm because it's easy and provides at least some guidance. It's also often a poor way to go because was not written to define you warrant and thus often doesn't have the objective standards against which you'll later want to judge your software.

Having said all this, have I ever had contracts where I've used some vague specifications as the benchmark for the warranty? I must confess that the answer is yes, I have.

The reasons are actually quite simple — it's all about time and money. Quite frequently, the parties don't want to take the time or spend the money necessary to prepare a more meaningful specifications document that was designed to serve as a benchmark for the warranty. We all know what a car should do. There's no similar common sense benchmark by which we can judge IT work. If you don't specify things in your contract, you may find yourself staring into the face of a bad situation.

IT contracting is often a time consuming and resources draining project. It takes effort to negotiate a deal and develop things like performance standards and acceptance testing procedures.

You could choose to march ahead with a contract that has as much definitive and clear material as a politician's stump speech. If all goes well, you'll feel like you made the correct judgment. In fact, you were just lucky. This method is okay for the desperate and those who like to play in the dark.

Who Takes the Loss?
When things go wrong in the world of IT, they can have far-reaching consequences. If your office network goes down, you have all the losses that go with the lack of productivity of your employees. If you're an car rental company and your reservation system crashes and burns, it's obvious how disastrous that could be. Who pays for these losses?

In IT, the answer is that the customer usually bears these risks. We can argue about how fair it is. The car rental company could say something like, "Let me get this straight. I pay you $5 million to update our reservation's software. It stops functioning, I lose millions of dollars and you think that you shouldn't be responsible for that loss?"

The standard IT answer to the question is "That's right!"

The developer's perspective may not be obvious to many customers, but it does ring of legitimacy in many cases. As the customer, you must build mission critical systems with enough redundancy and over capacity to prevent catastrophic mishaps.

The argument would be that it was the fault of the car rental company in our scenario, which decided that running a parallel reservation's network or providing for more capacity was too expensive.

Similar arguments are made when other IT disasters happen. If you lose data, the vendor says that you should have had better backups. I could create other examples, but the point remains the same. It's up to the purchaser of IT services to create enough redundancy to protect against unacceptable losses.

The vendor's answer to legitimately broken or poorly performing IT products is we’ll improve our "response time," in dealing with the issue, but we will never ever ever write you a check for your losses. You must largely accept this fundamental norm.

Yes, there are exceptions. I have seen and negotiated contracts with real teeth against vendors, but they are the exceptions.

So, as a buyer of IT goods and services, focus on what you can get. You should always negotiate for better response time guarantees than are first being offered. Follow this up by requesting specific escalation provisions, which helps to insure that if level one support can't get the job done in a reasonable amount of time, it will move up through the vendor's chain of command quickly. Good response time and escalation provisions can be worth their weight in gold when you're in crisis.

Some Red Flags--Damage Limitations

One of the things you should always focus on is damage limitations. Be leery of clauses like, “Vendor’s liability for any loss, damage or expense of any kind, resulting from the products or services, negligence, or any other cause whatsoever, regardless of the form of action, whether in tort or in contract, shall be limited to the selling price of
the products or services.” Variations on this type of clause may limit you to six months of service charges or some predetermined, and usually low, dollar figure.

Limitations of liability are negotiable and since a one sided damage limitation could emasculate much of what you gained in other aspects of your negotiation, I would suggest that you must focus on damage limits.

Here are some tips that can to some extent function as a checklist for you.

-Make the damage limitation mutual. What’s good for the goose should be good for the gander. There is no justification for a limitation of liability that only benefits the vendor.

-Exclude third party indemnity claims from the limitation. If a third party sues your company due to your vendor’s misconduct, the limitation of liability should not limit your indemnity claim.

-Seek more than a refund as the damage limit. It just seems fundamentally unfair that the vendor has no skin in the game beyond a refund.

-Exclude willful repudiation of the contract from the damage limit. This is designed to prevent the vendor from repudiating your contract and damaging you because the vendor acquired a more lucrative customer. In that scenario, you really should be getting at least your expectation damages.

Exclude breaches of the confidentiality provisions from the limitation of liability. I can just imagine a scenario where your confidential information is more valuable than the damage cap so they thoughtfully decide that theft of your information is profitable.

This checklist could be an article in itself, but the goal is this piece is just to provide an overview for negotiating tech deals.

A Final Tip From the Trenches

In negotiating your agreements, you must avoid that natural tendency to see the deal’s starting point as being the vendor’s form. You should first see the deal from your one-sided perspective. What do you want and need?

In a negotiation, you’re not likely to get everything you want either, but you must work to pull contracts back to the middle, i.e., back to what’s fair. You shouldn’t ask for changes in a vendor’s form only after asking yourself whether the change is significant. If it’s one sided in favor of the vendor, ask that the provision be made neutral.

If the vendor asks that you indemnify them for your wrongdoing, you should ask that they indemnify you for their wrongdoing. If they get attorney’s fees if they’re the prevailing party, then you should if you’re the prevailing party. If they can terminate the
agreement if you sell your company, the reverse should be true.

After you’ve put every unfair provision on the table, you can use the issue of “significance” to decide which points to give up. Certainly, not every point has equal importance to you.

Just remember, what’s good for them is good for you. That’s fairness.

Don’t walk into a deal thinking about how big they are. They want your business or they wouldn’t be talking to you. Sure, the Microsofts of the world budge less than the local vendor down the road, but they all bend. The only way to find out how far is to push back.