Technology Transactions

How to Negotiate, Draft and Close Better Deals
Copyright 2018. John A. Newman. All Rights Reserved. The original illustrations for this book are by Paula Paulson.

This book is provided for informational purposes only and is not offered in lieu of legal research or analysis by an attorney. This book is provided “as is” and “with all faults”.

The author, John A. Newman, is a graduate of Georgetown University Law Center (LLM), Wake Forest Law School (JD). He helps clients structure, draft and negotiate technology transactions, create model templates, structure new business models and resolve disputes.

He can be contacted at legal@aol.com
Introduction

This book focuses on practical legal skills needed to succeed in structuring, negotiating and documenting commercial technology transactions. The emphasis is on Virginia’s Uniform Computer Information Transactions Act (Va. Code Title 59.1 (Trade & Commerce), Chapter 43 (UCITA), Sections 501.1 through 509.2), with cross-reference to the Model UCITA and Model Comments. Many of the principles contained in UCITA have been adopted by common law in other jurisdictions, so this program should be of fairly general application.¹

Lawyers who handle technology transactions know there’s nothing “soft” about the software industry. It’s a tough business environment populated by an assortment of large and small players, each having competitive advantages, talents, hopes, aspirations and (in some cases) predatory instincts, fueled by fantastic economic opportunities.

It is not uncommon for a solo software consultant with no legal representation to negotiate an agreement with a global software company’s experienced attorneys. And it is not uncommon for a large company with no software experience to

¹ UCITA elicited heated debate during the 1990’s and early 2000’s and has only been adopted in Virginia and Maryland. Other states apply the common law or UCC Article 2 (Sale of Goods). Prof. Ray Nimmer observes that if the true test of a contract law statute lies in how little litigation it creates, UCITA is succeeding: after more than a decade, there is only a trickle of cases, virtually none in Virginia. See also, Hillman & Rachlinkski, Standard-Form Contracting in the Electronic Age, 77 N.Y.U. L.Rev. 429, 491 (2002) (“[W]e contend that UCITA maintains the contextual, balanced approach to standard terms that can be found in the paper world.”).
negotiate an agreement with an experienced software boutique.

To make it on either side of the table, you need to know something about technology transactions law. Better yet, you need to know how business, law and technology mix together into an ever-changing landscape of business models and deal-making.

Technology law developed as lawyers scrambled to help innovators capture opportunities or avoid legal problems. This book will give you practical tools for structuring, negotiating and enforcing good contracts in the software and technology industry.

Study this book and the next time you are presented with a potential deal, you will be in a better position to keep your wits about you, understand the issues and negotiate effectively.

Ignore these lessons and the “invisible hand” of the free market will often push deals closed, but on terms that too often spring back to haunt the participants (see next page...)

Sincerely,

John Newman, Esq.
“Remember Meee? I’m Last Year’s Bad Deal!”
# TABLE OF CONTENTS

## PART ONE: SOFTWARE DEVELOPMENT AND LICENSE AGREEMENTS

1. **THE NEGOTIATING AND CONTRACTING PROCESS**

   A. Setting the Stage for Successful Negotiations: What Makes People Tick?  
      1. Reciprocity  
      2. Commitment & Consistency  
      3. Social Proof  
      4. Liking  
      5. Authority  
      6. Scarcity

   B. Preliminary Mechanisms Affect Deal Outcomes  
      1. Review Your Marketing Literature  
      2. About Form Agreements  
      3. About Custom Negotiated Deals
4. Engagement Letters 9
5. Non-Disclosure Agreements 11
6. Teaming Agreements 13
7. Linking Web Documents into the Agreement 14
8. Easy Contract Formation with Clickwrap Agreements 15

II. ISSUES IN DEVELOPMENT & LICENSING TRANSACTIONS 18

A. Project Staffing Issues 18
B. How Good Software Design Affects Legal Issues 20
C. Progress Reports 21
D. Testing and Acceptance 22
E. Understand Your Pricing Dynamics 23
F. Cash Flow Matters 26
   • Late Payment Syndrome
   • Trickle Down Theory
   • International Issues
   • Revenue Recognition

vii
G. Ownership of Work Product

1. Time of Conveyance

2. About Reusable Code Libraries

   (a) Code Licensed from Library for Customer's Project

   (b) Code Licensed into the Library for Future Projects

H. Granting Exclusive Marketing Licenses

1. Geographical or Vertical Market Limits

2. Minimum Performance

3. Channel Conflict

4. Support Issues

5. Minimum End-User Terms

6. Electronic Restraints

I. Confidentiality

J. Warranties

1. Non-Infringement Warranty
2. Express Warranties 37
   (a) "Casual" Express Warranties 37
       • Review Advertising Collateral
       • Watch Post-Signing Reassurances
       • Beta Versions
   (b) Formal Express Warranties 39
   (c) Express Warranty: 508 Accessibility 39
3. The Implied Warranties 41
   • Merchantability of Software Programs
   • Informational Content
   • Licensee's Particular Purpose
   • System Integration
4. Disclaiming/Modifying the Implied Warranties 43
5. Invalidation of Warranties by Software Code Modifications 45
K. Overview of Liabilities 47
   1. Contractor Damages 47
   2. Customer Damages 47
L. Limitation of Liabilities
1. Limitation of Liability Clause
2. Compartmentalize Your Liability

M. Remedies
1. Limitation of Remedies
2. Failure of Essential Purpose
3. No Rewriting of Agreements
4. Carve-Outs for Certain Liabilities
5. The Right To Cure

N. Indemnities
1. Common Types of Indemnities
   (a) Non-Infringement Indemnity
   (b) Bodily Injury, Death & Tangible Property Damage
   (c) Data Breach, Confidentiality
   (d) General Indemnity
2. Coordination of Indemnities with Limitation of Liabilities

x
O. Assignments & Transfers of Contracts 51
   1. "No Assignment" Clauses 51
   2. Absent an Agreement 59

P. Electronic Regulation of Access, Use and Repossession 59
   1. Online Access Contracts 60
   2. Electronic Usage Restrictions 60
   3. Self-Help Repossession on Cancellation for Breach 63
      (a) Physical Repossession 63
      (b) Electronic Repossession 63

Q. Entire Agreement Clause 67

III. IMPORTANT ISSUES FOR INTERNATIONAL TRANSACTIONS 68

A. IP Protection 68

B. Governing Law & Forum Clauses 68
   1. Governing Law 68
2. Governing Forum 70
3. Arbitration Provisions 71
4. Export Restrictions 72
5. Foreign Corrupt Practices Act 72

PART TWO: SOFTWARE SUPPORT AGREEMENTS 74

A. Software Support Agreements Under UCITA 76

B. Understanding Different Types of Software Support Transactions 80
   1. Packaged Off-the-Shelf Software 80
   2. Packaged Core Software with Customizations 82
      (a) Core Software or Customized Software? 82
      (b) Customizations by Licensor or Consultant? 83
   3. Support for Custom Developed Software 87
      (a) Project Staffing Issues 87
      (b) How Good Software Design Affects Support 88
(c) How Testing and Acceptance Effect Support  89

(d) How Confidentiality Restrictions Affect Support  90

C. Other Key Software Support Issues  91

1. What is an "Error" in the Software?  91

2. Know the Types of Software Releases  92

3. Understand the Different Levels of Support  93

4. Service Hours of Support  94

5. Classification of Support Calls  95

6. Some Typical Support Case Classifications  95

7. The Importance of Defining "Response Times"  96

8. Using Escalation Procedures to Attract Needed Attention  96

9. Certain Licensee Responsibilities  98

10. Service Level Credits  99

11. Minimum Period of Support  100
D. How Revenue Recognition Rules Affect Support Contracts
   1. License Fee Revenue Recognition
   2. Post-Contract Support Revenue Recognition

PART THREE: OPEN SOURCE SOFTWARE

I. What is Open Source Software?

II. Open Source Software Policies

III. Definitions to Know

IV. Identifying the Governing OSS License & Relevant Legal Issues
   A. OSS License Classification
   B. Distribution of Derivative Works
   C. Notice and Attribution Responsibilities
   D. Software Warranty & Support Considerations
   E. The OSS Project Sponsor
   F. Record Keeping and Code Management
G. Compliance with OSS Licenses................. 118
H. Employee Communication, Education and Training................. 118
I. Software Developed by Vendors................. 118
J. Patent Considerations................. 119
K. Audits................. 119
L. Violations................. 119
M. Exceptions to Policy................. 119
N. Revisions................. 119
O. Roles and Responsibilities................. 119
P. Opportunities and Traps in Dealing with Open Source................. 120

EPILOGUE................. 121
PART ONE: SOFTWARE DEVELOPMENT AND LICENSE AGREEMENTS.

UCITA applies to “computer information transactions” which are agreements to “create, modify, transfer or license computer information or informational rights in computer information.” It includes computer software and data, and support contracts that relate to software and online systems. It does not include contracts relating to financial services, insurance services, motion pictures (except mass market license transactions) nor musical works or contracts for employment (other than independent contractors who are not journalists). Other exceptions apply. See Va. Code 59.1-501.3 (Scope; Exclusions).

I. THE NEGOTIATING AND CONTRACTING PROCESS

The root of all substance is process. To negotiate a software development or license agreement containing substantive provisions that achieve your business objectives, while protecting you legally, think consciously about ways to control the contracting process from the outset.

Click-Wrap Procedural Win: We are all familiar with Click-Wrap Agreements, which are presented on screen and must be accepted before completing the software installation or online access to the software service. Many vendors attempt to hide behind these click-wraps as a procedural barrier to having to negotiate their terms. It standardizes their contracts and their risk profile and eliminates legal negotiation fees. If their click-wrap is reasonable and addresses issues important to both parties, this may be a viable approach. If not, and the license fees are significant or
if the customer data to be loaded onto the system is sensitive, then you'll need to break through this barrier and negotiate at least a custom Addendum addressing your key issues.

Assuming an actual negotiation takes place, you will want to have an appreciation of “negotiating psychology” as well as early-stage mechanisms that can greatly affect the final outcome. While the subject of “what makes people tick?” in the context of legal negotiations merits an entire MCLE presentation, below are six tips to consider.

A. Setting the Stage for Successful Negotiations: What Makes People Tick?

A successful contract negotiation involves not only understanding your client’s goals and deal “drivers” but also understanding the other party and what drives their behavior. Simply ask direct questions and you are likely to get some surprisingly direct answers: “How does your sales compensation program work and where are you in the sales cycle?” “When do you really need to close the deal to recognize the revenue this period?”

In addition to explicit deal drivers, research has identified six mental models that are fairly universal as learned behavior.³

---

Applying these principles can help trigger a propensity in your counterpart toward automatic subconscious behavior that results in affirmative agreement (triggering the “Yes” response):

1. **Reciprocity.** The desire to reciprocate appears to cross over all cultures. Small favors build up equity that greatly improves the chances of future reciprocation. Giving candy or a mint increases a waiter’s tip. Giving $5 with each survey request is more effective than giving $50 for each returned survey. Including pre-printed gummed address labels in unsolicited fundraising requests improves the success rate from 18% to 35%. If you ask someone to purchase an expensive item and they refuse, ask them to purchase a less expensive item and the success rate will triple on the smaller sale (they perceive the secondary request as a concession that invites reciprocation).

**Lesson:** If you invite a potential contracting partner to

---

4 This phenomenon may result from the presence of mirror neurons in the brain. The healthy brain contains neurons that allow us to mirror the brain activity of other individuals. This common human capacity allows us to stand in another’s shoes and see their point of view. This explains, in neurological terms, our ability to empathize with others. It explains why storytellers, orators, singers, dancers, actors and politicians can connect and move an audience emotionally. It explains why lawyers who take time to hear out the other side can establish the mutual rapport that can lead to creative solutions and resolution of legal disputes. Is it a coincidence that every major religion (Hinduism, Judaism, Zoroastrianism, Buddhism, Confucianism, Islam, Bahat and Christianity) contains some close variation of the empathetic Golden Rule: “Do unto others as you would have them do unto you”?
a meeting, it’s helpful to provide refreshments, snacks and other small favors and to be as courteous as possible, even if (or especially if) the negotiation is expected to be contentious at times.

2. **Commitment and Consistency.** Once a person decides to back an idea, they tend to stick with it later. For example, ask someone to predict if they will vote on election day and the show-up rate increases by 700%. “How are you tonight?” (90% say “fine”; this affirmative response doubles the acceptance rate on solicitations). Similarly, it’s best to obtain a new client even for a very small order. They now see themselves as a customer. Commitments made publicly stick more than commitments made in private. Those that entail sacrifice are stronger (this is why hazing cements commitment). Getting commitment up front (volunteer on election day) keeps people on board even if they find out later it entails more sacrifice than expected (you need to show up at 5:00 am).

    **Lesson:** it is best to be explicit during negotiations and to peg you concession with a corresponding concession from the other side: “Do I understand correctly that if I can get you this discount price, you can close the deal by Friday?”

3. **Social Proof.** We take social cues from others in deciding how we will act, especially in an uncertain environment. Leaders are more effective when they have a cadre of committed followers, who in turn influence others (and so on). This is why bartenders “salt” the tip jar, as it suggests others are tipping and so should you. People learn
quickly by mimicking their peers (e.g., the Montessori method).

**Lesson:** This principle is exemplified by the phrase, “Nobody ever got fired by going with IBM” and by vendor websites that display logos of their key customers, to create a “me too” atmosphere of comfort.

4. **Liking.** Clarence Darrow said the main thing a trial lawyer should do is have the jury like his client. That is also true in negotiations: people tend to say yes to people they know and like. Tupperware’s business model rides on top of personal friendships, where participants feel almost compelled to purchase in order to support the friend’s business venture. The quality of the product is of secondary importance. In deciding whether to like someone, people tend to look at personal attractiveness, good grooming, nice clothes that mirror others in the group. Sincere compliments also help.

**Lesson:** It is always helpful before a meeting starts to chit-chat with the other party and their attorney about any sort of common interest, to establish some personal rapport, to defuse a situation with a little self-deprecating humor. It is often helpful to let the other party talk for so long as they want, to vent their frustrations. You want to hear what the other party has to say, what’s important to *them*. If you can, see if you can say “yes” several times in a row, not to signal that you necessarily agree with them, but simply that you understand their point of view. If you take time to penetrate their positions to understand the interests they are trying to advance, you will have a better chance of arriving at a mutually acceptable solution.
5. Authority. People tend to follow directions from an authority figure. If you convince a group leader of something, he/she will use chain of command to help obtain cooperation of others (persuade the decision-maker). Evidence of “apparent” authority comes from nice clothes, trappings of power (nice watch, car), association with prestigious institutions. Sly sincerity works on this principle (e.g., a waiter who advises a patron that something they ordered was not the best item on the menu, and recommends something better at a lower price, establishes authority and sincerity and gets better tips).

Lesson: dress conservatively in an understated but powerful manner, then after a reasonable time, break the tension by figuratively or literally rolling up your sleeves to signal a working session among professionals. Use layered authority to defer harder decisions to an “authority figure” who is not present (but who “generally follows my advice.”). From your perspective, try to get the authority figures (decision makers) on the other side directly involved. Caveat: exhibiting too much authority, especially by large law firms, can backfire if the other lawyer views it as superiority or arrogance. This can fuel a competitive reaction or an undercurrent of resentment that can be counterproductive to the client.

6. Scarcity. Make something scarce and hard-to-get, and it will increase demand. Research shows that cookies sell faster when there are only two cookies left, than when the jar is chocked full of cookies. This can be difficult in software sales, because the marginal cost of producing another copy of software is exactly zero. So look for other ways to create
scarcity, such as the need to make a purchase now to schedule training slots or implementation services (which, of course, are filling up fast....)

**Side-note:** After I read the *Influence* book, I decided to try it out. I had previously put an ad in Craigslist to sell my Classic Corvette Stingray, but received little response. I then modified the ad to include the following “scarcity” statement: “*If you’re going to buy a C3 (third generation Corvette) be sure to buy a 1972: that’s the last year that had metal bumpers on both ends and no pollution control gear to limit performance. There are only a few 1972 Corvettes left in the country. And when they are gone, that’s it. They aren’t making them any more.*” I sold the car to the next prospect for my full asking price. Coincidence?

Here are some other examples of the scarcity principle at work that are typically used by salespersons:

- “Unfortunately, I don’t think we have any more of these in stock. But if I can find one in the back, do I understand that you want to purchase it at this price?”

- “You might buy extra products, because I’ve just heard on the grapevine that production is backed up and I’m not sure when we will be getting any more.”

- “Exclusive limited time offer ends soon!”

Automatic behavior is most often triggered when one or more of the six principles are encountered and we are rushed for
time, stressed, uncertain, indifferent, distracted or fatigued. Automatic behavior saves us the mental effort of having to think consciously when we are already maxed out.

B. Preliminary Mechanisms that Affect Deal Outcomes.

1. Review Your Marketing Literature: Customer expectations can be established at unrealistically high levels if the lawyer does not review product literature, web site content and proposals. Watch for superlatives and overstatements of product capabilities. If you join a negotiation after such materials were delivered, be sure the resulting contract does not incorporate the proposal, but instead supersedes it with a SOW and contains a merger clause. If you’re the Customer, by contrast, have the agreement refer to the proposal or even incorporate it into the contract (see discussion of Express and Implied Warranties, below).

2. About Form Agreements: Taking the initiative procedurally may be as simple as putting your form agreement on the table first and using that as the starting point in negotiations. This strategy can work for licensing off-the-shelf software when your business model is mature and well-defined. The form agreement should protect your interests, but you should consider including other provisions important to the other side. If the form agreement is too one-sided and you don’t have the bargaining power to make it stick, the other party will simply disengage and insist on using its own form agreement.

3. About Custom Negotiated Deals: In a custom negotiated transaction, be careful about trying to start with an agreement from a prior deal that does not fit the situation.
Instead, use a top-down structured process to design an agreement to which both sides can make a legal and emotional commitment. Start by quizzing the business and technical people about each party’s objectives and concerns going into the deal. From this high level, produce a term sheet that identifies and deals with the structural issues conceptually. (later, volunteer to produce the first draft of the agreement).

**Observation:** Think carefully before signing a term sheet or a letter of intent. They can be binding enough to get you sued, but not sufficiently detailed to provide legal protection. State explicitly in the term sheet that it is non-binding and any definitive agreement must be executed by authorized representatives of both parties. If you are re-negotiating a “bad deal” it is also helpful to state that all communications and negotiations, whether written or oral, are settlement discussions not admissible into evidence.

4. **Engagement Letters:** Consultants often want to get their people on-site quickly with some initial consulting or a requirements analysis. Use a simple two-page Engagement Letter to document the key terms: the parties, scope of work, pricing and payment, ownership of work product, confidentiality, warranties/disclaimers, liability and remedies and some general terms (merger clause, amendments, choice of law, etc.). Engagement Letters differ from term sheets because they contain actual contract provisions, in a simpler, plain-English format. But you should only use Engagement Letters for small dollar projects or phases, since larger ones put more pressure on each provision of the agreement. No matter how busy you all are, be sure these “small” agreements get proper legal review.
“TGIF, Only Two More Working Days ‘Til Monday.”
5. **Non-Disclosure Agreements:** NDAs can be dangerous if they limit your R&D and future competition:

"Acknowledgment of Certain Competitive Projects. The Company acknowledges that Bigco may be engaged in the development of technology that may compete with existing or prospective technology or business plans of the Company, and that Bigco may also be discussing, or may discuss in the future, possible business ventures with competitors of the Company. Nothing in this Agreement or in the limited use of Confidential Information by Bigco hereunder will be deemed inherently to involve the disclosure or misuse of the Company's Confidential Information in violation of this Agreement. Bigco will be deemed in compliance with this Agreement if it uses due diligence to ensure that the individual employees or Representatives of Bigco who receive Confidential Information with the consent of the Company do not disclose such information to third parties in violation of this Agreement and do not disclose such Confidential Information to other personnel at Bigco that are not approved by the Company. Bigco will identify the name and position of each individual who receives Confidential Information and let the Company determine whether it wishes to permit such individual to obtain access to such Confidential Information.

Reservation of Rights. Nothing in this Agreement obligates a party to enter into any business transaction or relationship with the other party and nothing herein shall be deemed to preclude a party from entering into a business transaction with any other party."

Another issue seen in NDAs is a “Residual Rights” clause in which the recipient is free to reuse confidential information without restriction to the extent it is “retained in recipient’s unaided memory.” The logical fallacy of that clause is that your trade secrets are only protected against someone who can't remember them. Probably not what you intended....
The Law of Unintended Consequences
6. **Teaming Agreements**: these are used early in the contracting process to coordinate marketing efforts between two parties working on a joint proposal to a prospect. Decide whether the team members are exclusive to each other, specify the point of contact for customer negotiations, address confidentiality issues and disclaim consequential damages. And make sure the document contains an independent contractor clause and doesn’t refer to the team members as “partners” (as that implies a general partnership with joint and several liability). The proposed sub-contractor should also avoid committing in advance to a flow-down of the customer’s standard contract terms, particularly if they have not yet been seen. Instead, reserve the right to negotiate reasonable changes in the subcontract consistent with your part of the proposal. To the extent you are contributing off-the-shelf software, try to have your standard license agreement apply (customer T&C’s, which you may not have seen yet, often obtain broad licensing rights, source code rights, warranty, remedy and liability provisions that you will want to negotiate). And, if possible, specify explicitly whether the purchase/sale aspect of the agreement is binding:

**Observation**: Teaming Agreements may or may not be binding contracts for the sale of goods or services between members. In *W.J. Schafer Assoc., Inc. v. Cordant, Inc.*, 254 Va. 514, 493 S.E.2d 512 (1997), the agreement was considered too indefinite to be binding because it: (a) allowed the Prime to designate a replacement Subcontractor; (b) did not firmly commit the Sub to sell the product (which was still in development); (c) did not specify the purchase price in
advance (although the Sub later bid a price in the Proposal and adjusted it in the BAFO) and (d) there was no assurance that the product would exist if the prime contract was awarded. See also Cyberlock Consulting v. Info Experts, 939 F. Supp. 2d 572 (E.D.Va. 2013), aff’d 549 Fed. Appx. 211 (4th Cir. 2014) (teaming agreement was an unenforceable “agreement to agree” and subject to successful future negotiation of the subcontract); accord, Navar v. Fed. Bus. Council, 291 Va. 338 (2016). One way to improve the enforceability of a teaming agreement is to pre-negotiate the terms of the actual subcontract and SOW and attach it as an exhibit to the teaming agreement. The teaming agreement should recite that the parties’ agreement now exists and that it is not subject to future negotiation and will be executed by the parties if the team’s proposal wins the competition.

**Tip:** When the other party stubbornly clings to a one-sided clause, find a reason to make the clause reciprocal. This forces them to see the issue both ways and can soften their position.

7. **Linking Web Documents into the Agreement.**

Software vendors often incorporate by reference into the Agreement documents that are posted on the web. These may include Privacy Policies, Security Policies, Travel Policies, Acceptable Use Policies and so forth. It could also include product specifications, software documentation and price lists. It is important to review those documents, as they often contain “zingers” that define what your client is actually getting for the price. Those linked provisions may also provide
that they may be modified by posting a new version online. The power to modify the Agreement unilaterally means your client loses control over the terms and substance of the relationship. This is especially true if the linked document is effective upon posting a new version (without actual notice or consent) and does not limit the nature or extent of the possible revisions.

**Tip:** If it’s necessary to incorporate a linked document into the Agreement, it’s best to indicate that if future modifications are *materially adverse* to your client, that your client’s express consent is required, or that your client may terminate the Agreement for convenience and obtain a refund of any pre-paid fees. If that fails, you could indicate that changes to the linked document will not be allowed to modify certain provisions of the Agreement without a signed amendment (e.g., warranty, liability, indemnity, confidentiality, etc.). A related issue, of course, is to indicate that Purchase Orders and similar transaction documents may be issued for administrative convenience, but do not become part of the agreement.

8. **Easy Contract Formation with Clickwrap Agreements.**

UCITA indicates that a “writing” requirement can be satisfied by a computer “record” and that any “signature” requirement is satisfied by an “authentication.” Va. Code 59.1-501.5. To “authenticate” means either to sign it or, with the intent to sign a record, to execute or adopt an electronic symbol, sound, message, or process referring to, attached to, including in, or logically associated or linked with, that record. Va. Code 59.1-501.2(a)(6).
UCITA thus validates standard click-wrap license agreements in online access contracts, in which the user has a meaningful opportunity to review the terms prior to contract formation, and manifests assent, for example, by clicking “I Agree.” See, Va. Code 59.1-501.13:1 (“Opportunity to Review”) and 59.1-501.12 (“Manifesting Assent”). It is enough to show that the user could not have accessed the system without first manifesting assent to the Agreement. Va. Code 59.1-501.12(d).

When mass-market software is delivered on tangible media at retail, the clickwrap agreement that is later presented during installation is enforceable if the buyer has a right to return the software for a refund if the proposed clickwrap agreement is rejected. Va. Code 59.1-501.13:1 (“Opportunity to Review”).

**Observation:** Contract formation issues often are litigated when an opportunistic competitor “free rides” on an established website by harvesting data from it and using that data in a competing website. When that happens, the binding effect of the online Terms of Use (which usually prohibit that type of conduct) become paramount in importance.

**Note:** In *Cvent v. Eventbrite*, 735 F.Supp.2d 927, 937 (E.D. Va. 2010) the vendor had posted its Terms of Use at the bottom of the web page and did not orchestrate a click-wrap acceptance procedure as a condition of use. The Court characterized this not as a click-wrap agreement, but as a more casual “browse-wrap” contract “buried at the bottom” of the website that did
not achieve contract formation on those terms. Based on the facts alleged in the complaint, the vendor’s contract claim that relied upon provisions of the TOU to prohibit data harvesting was dismissed on summary judgment.

**Lesson:** lawyers should pay attention to how a click-wrap agreement is actually presented for acceptance and to work with the technical folks to ensure the process is legally valid. Presenting it as part of the installation or sign-up process, and ensuring it must be “accepted” before proceeding, is the best method. Be sure the agreement is accessible later and can be downloaded or printed off by the user. If the user rejects the license after payment has been made, then you must offer a return/refund option.

**Wrinkle:** One Catch-22 that arises with contract formation in online agreements is the choice of law. If the contract specifies Virginia law, for example, but Virginia UCITA law must be relied upon to establish that a contract was formed on those terms in the first place, you can end up in an odd circular argument. Still, courts in other jurisdictions have enforced clickwrap agreements based on UCC Art. 2 or common law analysis. See Specht v. Netscape Communications Corp., 306 F.3d 17 (2d Cir. 2002); Davidson & Associates v. Internet Gateway, 334 F.Supp.2d 1164 (E.D.Mo. 2004); I.Lan Systems v. Netscout, 183 F.Supp.2d 328 (D.Mass. 2002).
II. ISSUES IN DEVELOPMENT & LICENSING TRANSACTIONS

Software Development Agreements should be structured to provide a contractual framework to advance the technical process of designing, developing, testing, implementing and supporting a software system that helps the customer achieve its actual business objectives. It should also provide flexibility for reasonable changes, off-ramps and protections if the deal sours. End-User licenses for off-the-shelf products should be standardized to the extent possible, to give predictability to the licensor’s business and pricing model.

**Tip:** If possible, customers should try to include a termination for convenience provision, under which they are free to terminate the relationship upon some period of written notice, without having to prove a material default by the other party. This exit-ramp can be helpful when confronted by a non-performing counterparty and the customer wants to terminate without any contention (willing to write off any money paid to date). The customer should pay the counterparty for work performed that complies with the contract (this is where an acceptance procedure as a condition of invoicing and payment helps). Any remaining dispute can be narrowed down to the final invoice, in most cases.

A. **Project Staffing Issues.** It is important for both parties’ technical people to get to know each other prior to awarding the contract. Sometimes, bringing in an outside developer intrudes on the turf of in-house technical people or other contractors who can undermine a project’s success. The
customer will want to review the resumes of contractor personnel to be assigned to the project and to assess their technical ability and experience with similar systems. Each party should specify a Project Coordinator and the key technical people. The agreement should include a non-solicitation clause to maintain workforce stability and should provide some level of reassurance regarding staffing stability and commitments, replacement/approval procedures, adherence to workplace safety, security and substance abuse policies.

**Learning Curves.** Losing or having to replace project managers or key technical personnel during a project can have a devastating effect on progress and quality of work. Developers who quit may have gotten in over their heads technically, so their departure may signal problems with the project. Replacing key staff may entail a steep learning curve for the new people before they become productive. If the contactor removes staff without customer’s consent, the contract may require a transition instructional period and deny or limit customer’s obligation to pay for the orientation time. The customer may also restrict reassignment of contractor personnel to projects with key competitors for a specified period of time. The customer will also want to be able to request that the contractor pull someone off the project, simply because of personality issues. The contractor may resist that, given their desire to control their work-force.

**Restrictive Covenants:** If you are hiring contract workers from a placement agency (a “body shop”) or have contractor workers on site, assume the workers
are bound by non-circumvention and non-solicitation clauses (even if your own contract with their employer does not restrict it). The quickest way to trigger a heated dispute is to hire away a worker without conferring with their employer or agency. Small placement agencies are particularly aggressive in defending their interests and it is important to respect the value they contribute. If you represent the agency, be sure to include a non-solicitation and no-hiring clause in the customer agreement, as poaching workers is a constant temptation.

**Use of Consultants:** UCITA Section 505 states that a party may perform its contractual duties through a subcontractor unless the contract prohibits subcontracting or the customer has a substantial interest in having the contractor perform the work (Va. Code 59.1-505.5). Note: a ban on subcontractors may be implied if the agreement’s confidentiality clause restricts access to “employees” and prohibits disclosure to third parties [Section 505, Comment 2]. This clause could be loosened by replacing references to contractor “employees” with contractor “personnel” and including consultants within that category, so long as they execute an NDA. The customer may want the right to approve subcontractors, and the contractor will then add: “which approval shall not unreasonably be withheld.”

**B. How Good Software Design Affects Legal Issues.** Lawyers need to understand that many development projects fail because the parties do not allocate sufficient time and resources to identify the customer’s requirements and
prepare functional and detailed technical designs (including interfaces between all systems) prior to actual coding. A robust design phase allows you to unhitch design from development and is the foundation for successful development and debugging phases. Having a good design also helps demarcate lines of responsibility between any co-development teams and defines the programming interfaces. These are essential if the developed system is to operate in a legacy environment or with off-the-shelf software.

**Structuring Tip:** Consider breaking the project into separate phases: requirements analysis, functional design, technical description, coding & testing, implementation, training, support. Each phase must be completed, reviewed, accepted and paid for before the next phase commences. This discipline can be enforced by using a Basic Ordering Agreement (BOA) under which individual Task Orders are issued for each project phase. The contractor’s need to prove itself on each Task Order before another one is issued imposes discipline and tends to keep contractors on their good behavior. The Task Order approach may also help the contractor if the agreement compartmentalizes contractor’s liability to the amount paid under each Task Order.

C. **Progress Reports.** Both parties should be sure not to let the developers go off on their own for an extended period of development, since they can go astray or get sidetracked on projects for other customers that may be in crisis mode. It is important to monitor progress weekly and monthly. For larger projects, it is helpful to have your technical people (or an independent consultant) conduct periodic code
review sessions. Contractors working on a T&M basis will want the weekly reviews to include sign-offs on weekly time sheets. This contemporaneous sign-off procedure identifies performance problems early and makes it hard for customers to hold up payments for acceptable work.

**Tip:** Customers prefer to make “time of the essence” on key deliverables, while contractors will push to make all schedules “reasonably accurate estimates, subject to revision once the scope of the project is better defined” and governed by a broadly drafted force majeure clause.

**D. Testing and Acceptance.** Testing should be conducted pursuant to a written test plan that is derived from the design documents. Each unit of code should be individually tested for functionality and reliability. The system should then be tested to see whether the units function together as an integrated system and with other systems. Efficient debugging usually depends on how well the system was designed from the outset, as a good design allows developers to isolate, identify and correct bugs quickly. This is why a robust design is so important to the development of a functional and reliable system.

As noted earlier, there are two basic approaches to drafting an acceptance clause:

- **Contractor version:** “The Software shall be deemed accepted if Customer does not notify Contractor of any material defects during the Test Period or, if Customer makes beneficial use of the Software, it shall conclusively be deemed accepted and any outstanding punchlist items shall be covered to the extent provided under Section _ ("Warranties").”
• **Customer version:** “The Software shall be accepted as conforming to the requirements of this Agreement only when Customer issues its written certification of acceptance. If Contractor fails within thirty (30) calendar days to correct punchlist items or the Software otherwise fails to comply with the requirements of the Test Plan within the designated time, Customer may: (a) reject the Software in its entirety and recover amounts paid hereunder; (b) issue a "partial acceptance” of the Software, with an equitable adjustment in the price to account for such deficiency; (c) conditionally accept the Software, while reserving its right to revoke acceptance if timely correction is not forthcoming, or (d) pursue whatever other remedies are available under this Agreement.”

E. **Understand Your Pricing Dynamics.** Different pricing structures impose different dynamics on a project. In the commercial marketplace, most projects are commissioned on a firm fixed price basis or a time and materials (T&M) basis.

• **Firm Fixed Pricing:** It is axiomatic in software development transactions that a firm fixed price creates an immediate tension over the scope of the project before the ink dries on the signatures. The underlying tension can be defused somewhat by separating the design phase from the development phase (as suggested above), and pricing each one separately. Even then, be sure to include Change Control procedures to impose a formal process on managing changes to the scope. This is important because project people will often enter into numerous verbal side deals that remain under the legal radar until the project is behind schedule. You also want to make sure the fixed price is matched to a clearly defined scope of work and, if you are the customer, that the vendor’s stated “assumptions” that went into the pricing are not so
broad that they shift too much project risk to the customer. For that same reason, review the Force Majeure clause to ensure it’s not so broad that it provides a ready escape hatch for any delay.

• **T&M Pricing.** For T&M projects, many customers will require a total dollar budget for each phase. Be sure to specify whether this “budget” is a non-binding estimate or, instead, a not-to-exceed billing limit (a tight billing limit will create dynamics similar to a fixed price deal). Customers will want to match billing rates not only to positions (e.g. “Sr. Developer”) but also to specific individuals assigned to the project. This keeps the Contractor from raising rates indirectly through its reclassification of assigned workers.

• **Out-of-Pockets.** Be sure the contract specifies which party is responsible for out-of-pocket travel and lodging expenses, as well as incidental expenses, such as software development hardware, software tools, communications access and the like. In a fixed price deal or a T&M project with a billing cap, the parties might specify a budgeted number of trips and include a procedure for approving additional travel under a Change Control clause. You may also want to reference a party’s Travel Policy which should specify in some detail what travel costs are acceptable and what documentation is required.
Hitched to a Fixed Price Contract
• **Auditing Records.** Contractor should be required to maintain adequate documentation to substantiate T&M charges and requests for out-of-pocket reimbursements. Contractors should use time cards for T&M work that are signed off by Customer’s project manager each week. Customer will want the right to audit such records during normal hours upon reasonable advance notice. Contractor will want the audit to be conducted at customer’s expense and only once a year.

**F. Cash Flow Matters.** Cash flow (or lack of it) is a common point of contention in software transactions and one which accrual basis Contractors often overlook until payment is long overdue. The Contractor will want the right to suspend performance or stop work if amounts are not paid on time. Customer will want to tie payments to delivery of milestones and to its formal written acceptance of deliverables. The Contractor will counter this by wanting “deemed acceptance” to occur if the Customer is beneficially using the deliverable for a certain time, such as 30 days (the Customer will then insist on “30 consecutive days”).

**Late Payment Syndrome.** Contractors must not let Customers get behind on invoices. Customers can use this to extract a never-ending series of concessions on the project scope (usually meting out payment on one long overdue invoice, just when a new one is added to the stack). Moreover, they can often hold up the last milestone payment indefinitely, effectively depriving Contractor of its profit margin on the deal. If initial payments are late, it may be advisable to involve the
lawyer directly and invoke the right to suspend meetings or other project level work, focusing all eyes on the receivable until it is paid.

**Trickle Down Theory:** Subcontractors, especially those licensing off-the-shelf products, should watch for how payments are structured. Small subcontractors can get squeezed financially if its invoices are to be paid within a stated number of days after the prime contractor gets paid. You may never get paid if the prime contractor runs into performance or financial trouble (they will almost always use available money to make payroll before paying you). Instead, insist on being paid no later than a certain deadline, such as 60 or 90 days after invoice. Make sure the deadline is specified as non-contingent on customer payments and reserve the right to stop work.

**Virginia Law:** “Pay-if-Paid” clauses, in which a prime contractor is not obligated to pay its subcontractor until the prime is paid by the customer, are valid under UCITA and Virginia case law. See Pilar Services, Inc. v. NCI Information Systems, 569 F.Supp.2d 563 (E.D.Va. 2008), applying UCITA, and citing Galloway Corp. v. S.B. Ballard Constr. Co., 250 Va. 493, 464 S.E.2d 349, 355 (1995) (a pay-if-paid clause is an absolute defense for the prime contractor).

**International Issues:** Be sure to specify what currency governs and what the currency conversion reference date will be. If not specified, one party may
time payments to push the risk of currency fluctuations onto the other party.

**Revenue Recognition Tip:** Ordinarily, if you sign a contract and deliver a product before the end of a period, you can generally recognize the initial license fee that period even if the invoice is payable after period-end (e.g., Net 30). However, if a significant portion of the contract price is deferred for more than 12 months, the presence of the extended payment term will probably defer recognition of the *initial license fee* until the invoice actually becomes due. TPA 5100.42, SOP 97-2 Para. 28-29 (some companies are more conservative and require deferral if payments extend 6+ months). For this reason, you might limit any extended payments and/or make initial license fees due immediately upon contract execution (rather than net 30 after receipt of invoice).

**Revenue Recognition Tip:** If you provide discounts on future product licenses in order to close an immediate sale, this may require you to defer a proportionate amount of the *initial license fee* corresponding to the discounts into subsequent periods. TPA 5100.51, SOP 97-2. Thus, granting future discounts to close a sale can be somewhat self-defeating, especially for sales people whose commissions are tied to revenue.

**G. Ownership of Work Product.** When an employee develops software within the scope of his job, the employer will own the copyright to the work even without a written agreement. The opposite is true for independent contractors and consultants developing custom software, even
if source code is delivered. When hiring a contractor, there must be a written agreement containing an assignment of ownership to customer or the contractor will likely retain ownership. If the contractor will retain ownership, the agreement should specify the scope of customer’s license and whether source code is to be delivered to the customer.

1. **Time of Conveyance.** Delivery of a copy—even source code -- does not convey any ownership of IP rights in the software. Even where ownership is to be conveyed to the customer, the agreement will control the time at which it is conveyed. If not otherwise specified, ownership will be conveyed simultaneously as the program is being developed (delivery of a copy is not necessary). Or, if the agreement so provides, ownership could convey only after the customer pays the price (e.g., “Subject to payment of the price [for each deliverable], Customer will own all right, title and interest...”) [UCITA Section 501, Comment 3; See Va. Code 59.1-505.1].

**Tip:** The time of conveyance of work product is important when a payment dispute erupts. The vendor will want to modify the default rule under Va. Code 59.1-505.1 by linking conveyance of ownership to final payment. The customer will want immediate conveyance as work progresses. This keeps an aggressive vendor from trying to extract Change Orders for “out of scope” work that was really in-scope. When parties argue over time of conveyance, what they are really arguing about is leverage on price and payment issues.
Worker Classification Issues Can Impact IP Ownership
2. **About Reusable Code Libraries.** Over time, contractors may build up libraries of reusable code. Code libraries save development time, reduce prices and/or improve profits by allowing contractor to contribute pre-existing Embedded Software to the project. If the customer owns the work product, Contractor may ask for a license to add code developed for this customer to its library and reuse that code on future unrelated projects. Customers worried about funding the development of their competitors will want to restrict the contractor from reusing the technology on projects for key competitors. The agreement may identify the technology or competitors definitionally or by name, or it may include a case-by-case approval procedure (contractors will want approval not to be unreasonably withheld, while customers will want approval to be in their sole discretion).

(a) **Code Licensed from the Library for use on Customer’s Project:**

*Example:* “Except as otherwise specifically set forth in the Statement of Work: (a) this Agreement conveys no ownership rights by Contractor to Customer with respect to any Embedded Software, and (b) Contractor hereby grants Customer a paid-up, perpetual, nonexclusive license to use any Embedded Software in [object code] form strictly as an integral part of, and in conjunction with the Custom Work Product and for no other purpose. This license to Embedded Software is transferable by Customer (provided Customer destroys all remaining copies) but may not be sublicensed without Contractor’s prior written consent [, which consent shall not unreasonably be withheld].
(b) Code Licensed into the Library for Use on Future Projects:

Example: “Contractor is granted a royalty-free, perpetual, nonexclusive license to reproduce, modify, use and sublicense the Custom Work Product [solely in object code form] in conjunction with other development projects of Contractor that are not competitive with the Customer’s intended use of the Software [, provided, that for each project where Contractor wishes to reuse Custom Work Product, it shall first notify Customer of the intended use and obtain Customer’s prior written permission [which permission shall not unreasonably be withheld]].

H. Granting Exclusive Marketing Licenses.

Granting exclusive marketing rights should be carefully structured to avoid channel conflict and to prevent a non-performing marketing partner from locking out other channel participants.

1. Geographical or Vertical Market Limits. It is common to parse out exclusivity arrangements geographically according to the location of (a) the marketing partner, or (b) prospective customers. The “location” of a party/prospect could be based on the location of operating facilities, management offices, corporate headquarters or the headquarters of any upstream parent. Likewise, exclusivity could be limited to a particular vertical market of potential users (worldwide or in a particular region). Where software is being translated and localized for a particular foreign market, exclusivity can also be granted over the localized version.

2. Minimum Performance. Exclusivity is often conditioned upon the marketer producing a marketing plan,
committing a specified marketing budget and achieving a level of success (e.g., unit or dollar sales), perhaps after an initial ramp-up period. Failure to achieve these goals may convert the license to a non-exclusive basis. Product discount schedules may also be linked to sales levels.

3. **Channel Conflict.** Be sure any grant of exclusivity does not conflict with (a) a pre-existing exclusive or non-exclusive license covering the same products, (b) the same geographical scope, or (c) the ability to grant non-exclusive rights in target markets (e.g., an exclusive agreement in a specific target market may permit the granting of “global marketing rights” so long as the grantee is not headquartered in the target country). Be sure to coordinate with the grantor’s own marketing plans for direct sales. There may also be a “change of control” clause providing a right of termination if the marketer is acquired by a competitor of the grantor. Finally, the grantor may reserve the right to terminate the agreement on short notice or under specified conditions after a specified ramp-up period.

4. **Support Issues.** OEM agreements should specify which party is responsible for customer support and maintenance (distinguishing first level or “frontline” support from second level support). For example, it is not uncommon for the marketer to provide first level customer support directly with end-users of complex systems, since it may also be providing custom development, installation or training services. Marketers will sometimes want the right to continue providing support to end-users following termination of marketing rights.

5. **Minimum End-User Terms.** The grantor will want to ensure that the marketer either distributes products
containing grantor’s standard license terms or, if sublicensing is permitted, that the marketer’s contract with the end-user contains minimum terms and conditions protecting grantor’s IP rights and addressing warranty, liability and indemnity issues. If the marketer is providing risky consulting or implementation services, the licensor might distance itself by insisting that it End-User License Agreement be signed directly by the End-User (limiting the marketer’s contractual role to that of an order-taker and helping to ensure privity on limitation of liability protections).

6. **Electronic Restraints.** It is not uncommon in some third world markets for resellers to disregard the reseller agreement and distribute products without signed license agreements. There are also questions about local enforcement capabilities in some markets. Licensors selling into third world markets should consider using license keys, hardware dongles or other technology-based constraints as an additional enforcement mechanism (see discussion of passive electronic restraints, below).

I. **Confidentiality.** The confidentiality clause recognizes that in the course of a project, trade secrets and sensitive information may be disclosed that need to be protected. The clause may protect only the customer (especially if the customer owns all work product). Or it may be reciprocal, for example, if contractor retains ownership and licenses the software in source code form.

The confidentiality clause should restrict access, use and disclosure of Confidential Information (a defined term with carve-outs for pre-existing or independently developed knowledge or information in the public domain). It should also
impose a standard of care (e.g., “the same degree of care as recipient uses for its own confidential information, but in no event less than due diligence and care”) and should require the recipient to notify the disclosing party if its learns of any breach. A separate provision usually provides for injunctive relief to protect against any actual or threatened violation of the provision, without necessity of posting bond.

**Practice Tip:** Even if the contractor who retains ownership is willing to license source code, the customer will want the agreement to allow third party consultants to access and work with the code so long as they abide by written confidentiality agreements. Otherwise, the contractor may use the confidentiality clause to leverage out competitors and “sole source” ongoing support. Contractor may, however, justifiably want to preclude key competitors from gaining access to its source code under the guise of providing maintenance, so the agreement may exclude certain competitors definitionally and/or by name.

**Observation:** UCITA Section 807 states: “The remedy for breach of contract for disclosure or misuse of information that is a trade secret or in which the aggrieved party has a right of confidentiality includes as consequential damages compensation for the benefit obtained as a result of the breach.” (emphasis added). See Va. Code 59.1-508.7(c).
Daytime Partners, Nighttime Poachers
J. Warranties.

1. **Non-Infringement Warranty.** Customers expect either good title or at least quiet enjoyment of licensed technology free of any third party claims of infringement:

   **Example:** “Contractor represents and warrants [to the best of its knowledge and belief] that the Software, when properly used as contemplated herein, will not infringe or misappropriate any [United States] copyright, trademark, patent, or the trade secrets of any third persons [and shall defend, indemnify and hold Customer harmless from all costs, damages, liabilities, expenses (including reasonable legal fees) arising from any third party claim to the contrary.] Upon being notified of any claim contrary to such warranty, Contractor shall at its own expense: (i) defend through litigation or obtain through negotiation the right of Customer to continue using the Software; (ii) rework the Software so as to make it noninfringing while preserving the original functionality, or (iii) replace the Software with functionally equivalent software. If none of the foregoing alternatives provide an adequate remedy, Customer may terminate all or any part of this Agreement and recover amounts paid hereunder [with respect to the infringing Deliverable].

2. **Express Warranties.**

   (a) **“Casual” Express Warranties.** UCITA Section 402 (“Express Warranty”), Va. Code 59.1-504.2, states that an express warranty is created by, and that a deliverable will conform to:

   • An affirmation of fact or promise by the licensor to the licensee, including advertising, which relates to the licensed content and becomes a basis of the bargain;
• Any description, sample, model or demonstration of a **final product**, which is made part of the basis of the bargain (taking into account differences between the content and how it will be actually used). Note: Section 402 says product descriptions “will conform,” while samples, models and demos “will reasonably conform....”

**Review Advertising Collateral.** Commentary to Section 402 indicates that advertising (including web content) can become a basis for the bargain if it merely “played a role” in the decision to enter into the transaction. Commentary further indicates that once an express warranty is made, “the obligations created ordinarily cannot be easily deleted.” For this reason, lawyers should review marketing materials on a regular basis to prevent unintended express warranties.

**Watch Post-Signing Reassurances.** Contractors need to be careful about giving customers post-contract reassurances that may be at odds with the agreement. UCITA commentary states: “If language is used after the closing of the deal (as when the licensee on taking delivery asks for and receives an additional assurance), the assurance may become a modification of the contract. An agreed modification requires no consideration to be binding.” [UCITA Section 303(a); Section 402, Comment 2].

**Beta Versions.** Because the “sample, model or demonstration” that creates an express warranty
relates to a “final product,” the authors of UCITA recognize that beta versions are ordinarily understood not to be in final form and that the final deliverable will vary from the beta [UCITA Section 402, Comment 5].

(b) **Formal Express Warranties.** Contractors will expressly state in the agreement the extent to which they warrant their software. For example:

“Limited Warranties.

(i) **Limited Software Warranty.** Licensor warrants for a period of ninety (90) days following delivery of the Software that the Software will perform substantially in accordance with the Documentation.

(ii) **Limited Media Warranty.** Licensor warrants for a period of ninety (90) days following delivery of the Software that the CD-ROMs, diskettes or other media upon which the Software is delivered are free from defects in materials and workmanship under normal use.

(iii) **Limited Technical Support Services Warranty.** Licensor warrants for a period of ninety (90) days following performance of the service that its Technical Support Services will be performed consistent with generally accepted industry standards.”

(c) **Express Warranty of “508 Accessibility.”** Under Section 508 of the Federal Rehabilitation Act (29 U.S.C. 794(d)), software, websites and digital content (e.g., downloadable PDFs) should be accessible to handicapped users. Content developers should use contrasting colors (of a certain contrast ratio) so color-blind users can distinguish the features;
photos should be tagged with alternative textual descriptions so “screen readers” used by the blind can read them out loud; video should be accompanied by captions or subtitles so the deaf can understand them. This is likely to be an area of increased enforcement, as the DOJ will issue final regulations soon. Those regulations will incorporate features from an international technical standard, the “Web Content Accessibility Guidelines, Version 2.0” (known as “WCAG 2.0”). The guidelines promote digital content that is “perceivable, operable, understandable and robust.” The following contract clause should help address this issue:

“To the extent applicable, the [Software / Website / Deliverables] will be compliant with:

(A) Section 508 of the federal Rehabilitation Act (29 U.S.C. 794(d)) (“Section 508 Standards”), particularly the Web Content Accessibility Guidelines (WCAG 2.0, AA Standard), which requires electronic and information technology, including software applications, Internet Web sites and related content, to be accessible to people with disabilities; and

(B) any other applicable state or international law or regulation, over jurisdictions in which the Services or deliverables are to be provided, which require accessibility greater than the Section 508 Standards, and which are in effect at the time of contracting.”

**Practice Tip:** Accessibility complaints are most likely to be lodged by consumer advocacy law firms against a large retailer with a non-compliant website. An individual consumer may also lodge a complaint with DOJ (investigations are easy, since the websites are typically publicly available).
**Hedging Strategy:** If you are representing a licensee that wishes to use third party software or online services that are not yet compliant, then you will want to understand their roadmap for achieving compliance, weigh the risk of proceeding and include an indemnity under which the vendor will hold your client harmless from third party claims. Consider the vendor's financial ability to cover such risks (including insurance coverage). You might also include a right to terminate for convenience and recover pre-paid fees if such “accessibility” claims do arise.

3. **The Implied Warranties:** UCITA will imply various warranties under different circumstances unless disclaimed, including some new warranties:

- **Merchantability of Software Programs** (i.e., not data or informational content): a merchant impliedly warrants to **end-users** that the program is fit for its ordinary purpose and warrants to **distributors** that products are packaged and labeled as required and that the program conforms to any factual statements or promises appearing on the package and labels. This implies “fair average” or middle-of-the road quality. UCITA Sec. 403; Comment 3(a); Va. Code 59.1-504.3.

- **Informational Content** (i.e., not the software program itself): a merchant that is in a special relationship of reliance with a licensee, collects, compiles, processes, provides or transmits informational content warrants to that licensee that there is no inaccuracy in the content **caused by licensor’s failure to perform with reasonable care.**
This warranty does not apply to “published information content” or to someone who is a conduit or provides only editorial or summarization services relating to third party content. Reasonable care does not require perfect results, and the level of required care depends on the circumstances. UCITA Sec. 404; Va. Code 59.1-504.4. **Note:** this is a new implied warranty.

**Observation:** The merchant and customer must be in a relationship of special reliance for the information content warranty to apply: “A fundamental aspect of a special relationship is that the information provider is specifically aware of, and personally tailors information to the needs of the licensee.” It does not apply to information provided in standard format, but instead “must be personally tailored for the recipient.” The warranty does not apply to published information content, which is treated analogously to published books or newspapers.

- **Licensee’s Particular Purpose:** if the licensor has reason to know of the particular purpose for which the computer information is required, and that the licensee is relying on licensor’s skill or judgment to **select, develop or furnish** suitable information, there is an implied warranty that the information is fit for that particular purpose. The warranty covers software, content and documentation. Section 405; Va. Code 59.1-504.5.

**T&M Caveat:** if a developer is paid for the time or effort expended (a T&M deal), then the implied
warranty is that the information will not fail to achieve the particular purpose as a result of the licensor’s lack of reasonable effort.

- **System Integration:** if an agreement requires licensor to provide or select a system consisting of computer programs and goods (hardware) and the licensor has reason to know that the licensee is relying on the skill or judgment of the licensor to select the components of the system, there is an implied warranty that the components will function together as a system. Section 405; Va. Code 59.1-504.5(c).

**Example:** If the agreement provides that the licensor will select a computer, five software applications and a printer, the warranty is that the software will run on the computer and the printer will work with the software and computer [Comment 6].

4. **Disclaiming or Modifying the Implied Warranties.** Va. Code 59.1-504.6:

- **Short Form Disclaimer:** “AS-IS” and “WITH ALL FAULTS” is still effective to disclaim implied warranties, including those under UCITA (except the non-infringement warranty). As usual, the disclaimers must be CONSPICUOUS.

- **Long Form Disclaimers:**

  (a) **Warranty of Merchantability** is disclaimed by including magic words disclaiming “MERCHANTABILITY” or “QUALITY” or words of similar import;
(b) **Warranty of Informational Content** is disclaimed by including magic words disclaiming “ACCURACY” or *words of similar import*;

(c) **Warranties of Licensee’s Particular Purpose & System Integration** are disclaimed by statements such as “THERE IS NO WARRANTY THAT THIS INFORMATION, OUR EFFORTS OR THE SYSTEM WILL FULFILL ANY OF YOUR PARTICULAR NEEDS” or words of similar import.

**Saving Grace:** To “avoid a trap for the unwary,” UCITA makes clear that traditional disclaimers of “merchantability” and “fitness for a particular purpose” under UCC Article 2 and 2A are effective to disclaim the implied warranties described immediately above (including new ones) [UCITA Section 406(b)(4); Comment 4(e); Va. Code 59.1504.6(b)(4)].

**Example: Disclaimer of All Other Warranties.** THE WARRANTIES ABOVE IN THIS SECTION __ ARE EXCLUSIVE AND ARE IN LIEU OF ALL OTHER WARRANTIES, WHETHER EXPRESS OR IMPLIED, INCLUDING THE IMPLIED WARRANTIES OF MERCHANTABILITY, TITLE, ACCURACY, SYSTEM INTEGRATION, QUALITY AND FITNESS FOR A PARTICULAR PURPOSE. NO ORAL OR WRITTEN INFORMATION OR ADVICE GIVEN BY LICENSOR, ITS DEALERS, DISTRIBUTORS, AGENTS OR EMPLOYEES (INCLUDING ADVERTISING MATERIAL) SHALL CREATE A WARRANTY OR IN ANY WAY INCREASE THE SCOPE OF THE WARRANTIES GIVEN IN THIS SECTION, AND LICENSEE MAY NOT RELY ON ANY SUCH INFORMATION OR ADVICE.

Licensor does not warrant that the Software will meet Licensee’s requirements, that the Software will operate in combinations other than as specified in the Documentation, that the operation of the Software will be uninterrupted or error-free or that
Software errors will be corrected. Pre-production releases (including Alpha and Beta site releases) of Software and Technical Support Services related thereto are distributed “AS IS” and “WITH ALL FAULTS.”

**Customer Examination:** UCITA indicates that, “if a licensee before entering into a contract has examined the information or the sample or model as fully as it desired or has refused to examine the information, there is no implied warranty with regard to defects that an examination ought in the circumstances to have revealed to the licensee.” This can be relevant where the licensee is provided a copy of software or samples on a “try before you buy” basis [Section 406(d)]. This would not exclude an implied warranty against latent defects that passed initial review but surfaced later under more stressful actual operations.

5. **Invalidation of Warranties by Software Code Modifications.** If licensee modifies a software program, this will invalidate all express and implied performance-based warranties regarding the modified version. A modification occurs if the licensee adds code, modifies code or deletes code from the program. A modification does not occur if the licensee simply uses existing program features to configure software or make normal use a Toolkit to assemble programs from pre-existing components. No invalidation occurs if the parties are engaged in joint development and one party modifies the other’s program as part of its authorized scope of work [UCITA Section 407; Comments 1-2; Va. Code 59.1-504.7].
“Yes she’s fully warranted to be without faults or faculties.”
K. **Overview of Liabilities.** Absent an effective exclusion or limitation of damages, the following general rules apply:

1. **Contractor Damages:** UCITA Section 808, Va. Code 59.1-508.8, generally allows the licensor to recover: (a) direct damages equal to the price due under the agreement, fewer expenses avoided as a result of the breach (yielding lost profit and overhead) plus (b) incidental and consequential damages. The contractor has a general duty to mitigate loss. If the breach makes it possible for the contractor to enter into a “substitute transaction,” damages are reduced either by a reasonable actual substitute transaction fee or by a reasonable hypothetical substitute transaction.

   **Examples:** Comments to UCITA Section 808 indicate that if the licensee under a non-exclusive license fails to pay, the contractor may recover the total license fee due. There is no reduction for a “substitute transaction” on a non-exclusive license since contractor could have granted any number of similar licenses. However, if the first license was exclusive, then the recovery should be reduced by any actual or hypothetical license fees from a substitute exclusive license that was made possible by breach and cancellation of the first license. These rules do not affect separate recovery under IP law (e.g., if the licensee continues to use the product beyond the term or permitted scope).

2. **Customer Damages:** UCITA Section 809, Va. Code 59.1-508.9, generally allows a licensee to recover from a breaching
licensor: (a) direct damages upon breach equal to: (i) recovery of payments already made for the defective performance; (ii) the FMV of the performance less the contract fee or (iii) the reasonable cost of substitute performance ("cover") if obtained without undue delay and for a substantially similar product with the same scope of use terms, and (b) incidental and consequential damages (including lost profits).

Consequential damages may be limited by contract and are generally not recoverable under UCITA with respect to “published informational content” (e.g., online newspapers, interactive games, stock ticker data feeds and the like). Section 807(b)(1); Va. Code 59.1-508.7(b)(1).

Note: licensee should try to match the terms for any “cover” transaction to the original license granted (i.e., it’s best if licensee gets the same product with the same license terms from another distributor at a higher cost, if necessary, and then recover the price difference as cost of cover).

L. Limitation of Liabilities.

1. Limitation of Liability Clause. It is common in software transactions to limit liabilities for direct damages to a refund of the price paid (or perhaps to 12 months’ subscription fees for online transactions) and to disclaim liability for

---

indirect, incidental and consequential damages. The contactor will want to carve out exceptions for customer’s breach of IP related clauses and licensee will want an exception for contractor’s violation of the non-infringement warranty (you’ll need to coordinate those issues with the Indemnity clause, discussed further below):

Example: “NOTWITHSTANDING ANY PROVISION OF THIS AGREEMENT TO THE CONTRARY, EXCEPT FOR LICENSEE’S VIOLATION OF ARTICLE __ ("PERMITTED USE") OR SECTION __ ("CONFIDENTIALITY") [OR CONTRACTOR’S VIOLATION OF SECTION __ ("NON-INFRINGEMENT WARRANTY")]) HEREOF, IN NO EVENT SHALL EITHER PARTY, NOR ANYONE ELSE WHO HAS BEEN INVOLVED IN THE CREATION, PRODUCTION OR DELIVERY OF THE SOFTWARE, INCLUDING CONTRACTOR’S LICENSORS, BE LIABLE FOR ANY INDIRECT, INCIDENTAL, SPECIAL, PUNITIVE OR CONSEQUENTIAL DAMAGES, OR DAMAGES FOR LOSS OF PROFITS, REVENUE, DATA OR USE, INCURRED BY EITHER PARTY OR ANY THIRD PARTY, WHETHER IN AN ACTION IN CONTRACT OR TORT, EVEN IF SUCH PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

CONTRACTOR’S LIABILITY FOR DAMAGES AND EXPENSES HEREUNDER OR RELATING HERETO (WHETHER IN AN ACTION IN CONTRACT OR TORT) SHALL IN NO EVENT EXCEED THE AMOUNT OF LICENSE FEES PAID TO CONTRACTOR WITH RESPECT TO THIS AGREEMENT, AND IF SUCH DAMAGES RESULT FROM CUSTOMER’S USE OF PARTICULAR LICENSE FILES OR TECHNICAL SUPPORT SERVICES THEN SUCH LIABILITY SHALL BE LIMITED TO LICENSE FEES PAID TO CONTRACTOR FOR THE RELEVANT SOFTWARE OR SERVICES GIVING RISE TO THE LIABILITY. BECAUSE SOME STATES AND JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF LIABILITY, PORTIONS OF THE ABOVE LIMITATION MAY NOT APPLY TO YOU.
THE PROVISIONS OF THIS AGREEMENT ALLOCATE THE RISKS BETWEEN CONTRACTOR AND CUSTOMER. CONTRACTOR’S PRICING REFLECTS THIS ALLOCATION OF RISK AND THE LIMITATION OF LIABILITY SPECIFIED HEREIN.”

Watch the End Run on Liability: Agreements often contain a standard indemnity for negligent or willful acts causing bodily injury or property damage. Contractors will want to narrow these clauses to avoid opening up liability for computer malpractice. For example, damage to “property” should be limited to “tangible property” to avoid implying liability for damaged software or data. Also, try to make the indemnity subject to the overall limitation of liability provision or limit exposure to the amount of actual insurance coverage.

2. Compartmentalize Your Liability: Contractors supplying both off-the-shelf software and custom development or implementation services should consider using a separate Software License Agreement and a Software Development Agreement. This helps compartmentalize your liability. If your contract limits liability to a refund of the price paid, then a failure by the development team only risks the development money, without jeopardizing the license fees received. The customer, of course, will want to combine the provisions, or cross-reference the default and liability provisions to avoid such compartmentalization.

M. Remedies. UCITA Section 801, Va. Code 59.1-508.1, reflects the general rule that “remedies are cumulative, but a party may not recover more than once for the same loss.” It also makes clear that “the parties by agreement may alter a remedy or make it unavailable. The agreement governs unless
expressly invalidated by this Act.” [Comment 2]. UCITA also makes clear that a remedy for contract breach, such as exceeding the scope of use under a license, does not displace a remedy for copyright infringement: “often the two forms of recovery refer to different damages and are not a double recovery.” [Comment 4].

1. **Limitation of Remedies:** UCITA Section 803, Va. Code 59.1-508.3, generally allows the agreement to limit or alter the measure of damages. The agreement may also limit other remedies, such as by precluding a party’s right to cancel for breach of contract, limiting remedies to returning copies and refunding the price, or limiting remedies to repair or replacement of nonconforming copies. Resort to a contractual remedy is considered optional, unless the remedy is expressly agreed to be exclusive, in which case it is the sole remedy. The following sample clause illustrates the classic “repair, replace or refund” as the exclusive remedy:

   **Example:** “If the Software does not perform as warranted, Contractor shall undertake to correct the Software, or if correction of the Software is reasonably not possible, replace such Software free of charge with conforming software. If neither of the foregoing is commercially practicable, Contractor shall terminate this Agreement with respect to the non-conforming software program and refund the monies paid by Customer attributable to such non-conforming software program. The foregoing are Customer’s sole and exclusive remedies for breach of warranty.”

2. **Failure of Essential Purpose:** UCITA Section 803, Va. Code 59.1-508.3, also makes clear that “failure or unconscionability of an agreed exclusive or limited remedy makes a term disclaiming or limiting consequential or
incidental damages unenforceable unless the agreement expressly makes the disclaimer or limitation independent of the agreed remedy." Official comments to Section 803 indicate that the remedy and liability provisions will be deemed dependent unless the agreement expressly indicates otherwise. This is contrary to the rule under UCC Article 2.

**Interplay Between Remedy & Damage Limits:** Comments to Section 803 place great emphasis on whether a “repair or replacement” remedy is backed up by a right to a refund. Without a right to a refund, a contractual disclaimer of consequential damages will not be upheld unless it is drafted to be independent of the remedy provision. So it is good practice, if possible, to add a “refund” right to your exclusive “repair or replace” remedy (remember that refund rights are also central to the enforceability of shrink-wrap licenses). **Note:** refunds may be restricted to money paid for a defective product alone, without refunding money paid for other services or products that were not defective.

3. **No Rewriting of Agreements:** Commentary to UCITA also states: “Remedy terms are agreed allocations of risks,” and that “if a remedy is provided and is made exclusive, the fact that it does not fully compensate the aggrieved party is not a reason to allow that party to avoid the consequences of its agreement.”

**Example:** A contract limits recovery for software defects used in a satellite system to the price of the software ($100,000). A defect in the software causes
the satellite to fall out of orbit, destroying the $1 million satellite. The damage limit is valid. The decision to limit damages to $100,000 affects pricing and risk and cannot be set aside just because the loss fell mostly on one party. [Comment 6 to UCITA Section 803].

4. **Carve-Outs for Certain Liabilities:** If the consequential damage limit holds up, it will apply to all such loss incurred by the aggrieved party under the agreement, unless the agreement states otherwise. For this reason, a consequential damage limit often carves out an exception for claims relating to IP infringement, breach of confidentiality or exceeding the license scope. Section 803 also states that exclusion of liability of consequential damages for personal injury is presumed unconscionable, while exclusion of consequential damages for commercial loss is not. Va. Code 59.1-508.3(d).

5. **The Right To Cure.** The contractor generally has a right to cure a breach if it does so before any deadline for performance passes. It may also cure if it reasonably believes the performance would be acceptable (with or without a price adjustment) and it notifies the customer and proceeds to cure within a reasonable time after performance was due. Contractor may also cure if it notifies customer of its intentions and actually cures before customer cancels the contract [UCITA Section 703; Va. Code 59.1-507.3].
The Risk/Return Tradeoff
N. Indemnities

1. Common Types of Indemnities.

(a) Non-Infringement Indemnity. This indemnity allocates the risk of a third party infringement claim brought against either party as a result of licensee’s use of the software. Below is a fairly typical long-form indemnity clause (the phrase “Intellectual Property Rights” is a defined term that usually refers to [United States] copyright and trade secrets):

**Indemnity by Licensor.** Licensor will defend, indemnify and hold Licensee harmless from and against any loss, cost and expense that Licensee incurs because of a third party claim that the Subscription Software infringes any United States patent or other Intellectual Property Rights of others. Licensor's obligations under this indemnification are expressly conditioned on the following: (i) Licensee must promptly notify Licensor of any such claim; (ii) Licensee must in writing grant Licensor sole control of the defense of any such claim and of all negotiations for its settlement or compromise (if Licensee chooses to represent its own interests in any such action, Licensee may do so at its own expense, but such representation must not prejudice Licensor's right to control the defense of the claim and negotiate its settlement or compromise); (iii) Licensee must cooperate with Licensor to facilitate the settlement or defense of the claim. Licensor will not have any liability hereunder to the extent the claim arises from (a) any modification of the Subscription Software; or (b) the use or combination of the Subscription Software with any computer, computer platform, operating system and/or data base management system other than provided by Licensor. In addition, if any Subscription Software is, or in Licensor's opinion is likely to become, the subject of an
infringement claim, then Licensor, at its sole option and expense, will either: (A) obtain for Licensee the right to continue using the Subscription Software under the terms of this Agreement; (B) replace the Subscription Software with products that are substantially equivalent in function, or modify the Subscription Software so that it becomes non-infringing and substantially equivalent in function; or (C) refund to Licensee the un-used portion of the Subscription Services fee, if any, paid to Licensor for the Subscription Software giving rise to the infringement claim, and discontinue Licensee’s use of such Subscription Software. **THIS SECTION SETS FORTH LICENSOR’S EXCLUSIVE OBLIGATION AND LIABILITY WITH RESPECT TO INFRINGEMENT OF INTELLECTUAL PROPERTY RIGHTS.**

(b) **Bodily Injury, Death and Tangible Property Damage.** Both parties will probably want the other party to indemnify and hold them harmless from any bodily injury, death or property damage proximately caused by the other party. These are damages that are typically covered by insurance, but this contractual clause does not usually contain any deductibles or coverage limits of the type seen in insurance policies. As noted earlier, “property damage” is often limited to “tangible property” to avoid claims for loss of intangible data (which is personal property). The data backup procedures are usually relied upon as the best protection against data loss, which most software vendors typically disclaim.

(c) **Data Breach, Confidentiality.** If the vendor fails to follow good data and software security practices and customer data is lost to a data breach or a willful breach of the confidentiality protections, these claims are often subject to an indemnity.
Encryption Tip: The risk of data breach and loss of confidential information (particularly PII, or “personally identifiable information”) can be reduced through the use of encryption. It’s best to encrypt data during transmission (in transit) and at rest (in storage). And, if you are the customer using a cloud computing vendor, try to manage the encryption keys yourself. Cloud computing vendors prefer not having access to customer data, and key management is usually available as an option under their security plan. Using encryption and tight key management practices can greatly reduce the likelihood of a data breach by narrowing the scope of those who have practical access to such information.

Insurance Tip: Third party claims for data breach and privacy violations are regularly excluded from coverage by General Liability, E&O and Directors & Officers Liability Policies. The cost of a data breach investigation can be large. In one recent investigation, outside security forensic experts charged $150,000 for about 3-4 weeks of emergency consulting services (at $400 per hour). To cover this risk, be sure there is an “Internet Liability” policy in place ($1M seems typical). This covers cyber-liability (third party claims that arise from doing business digitally) and privacy liability (claims for data breach disclosing personally identifiable information (PII)). Your contract with a vendor should require naming your client as an additional insured and include a waiver of subrogation (to be reflected on the Certificate of Insurance (COI) given you). Waiver of subrogation means the insurance company can’t come after you if the covered loss was your fault. This waiver must occur prior to the actual loss, so it’s a pre-contract checklist item you
need to complete. Note: I’ve actually seen a fake COI provided by a vendor on one occasion, so you might call the insurance broker and verify the existence of coverage if the certificate did not originate directly from the broker.

(d) General Indemnity. Some model agreements will attempt to impose a general indemnity on the vendor for “any breach of this Agreement.” Those general indemnities should be vigorously resisted by counsel for the vendor. The best argument is simply that the price reflects the limited liability and allocation of risk in the model document. Resistance to this type of clause is particularly important if there is an exception to the limitation of liability clause for indemnified claims (see below).

2. Coordination of Indemnities with Limitation of Liabilities. The parties will want to negotiate over coordination of the various indemnities with the limitation of liability clause. When a customer is represented, the lawyer will want to negotiate an exception from the vendor’s limitation of liability clause for these types of claims (“Except for indemnified claims described in Section __, the Licensor is not liable for direct damages exceeding the price paid, nor for any indirect, incidental, special or consequential damages....[etc.]”

O. Assignments & Transfers of Contracts.

1. “No Assignment” Clauses. UCITA Section 503, Va. Code 59.1-505.3, provides that a restriction on transferring a party’s contractual interest is generally enforceable. In a mass-market transaction, the transfer restriction must be conspicuous. Note:
“mass-market” means retail level (i.e., primarily consumer) licenses of off-the-shelf products. This means you should update all your standard licenses to make any “no assignment” clauses **CONSPICUOUS**.

2. **Absent an Agreement.** In the absence of a provision restricting assignment, a party may generally transfer its contractual interest unless the transfer:

   (a) is prohibited by other law (e.g., this could be a big exception since courts have held that IP law prohibits transferring a non-exclusive copyright or patent license; see also 17 U.S.C. Section 107, prohibiting unauthorized software rentals);

   (b) would materially impair the other party’s interests, by:

   - changing the other party’s duty (e.g., assigning a company-wide support and maintenance agreement from a 20 man company to AT&T);
   - increasing the burden or risk imposed on the other party (e.g., assigning a source code license to a company located in Iraq);
   - impairing the other party’s property (e.g., assigning an enterprise-wide license granted to a 20 man company to AT&T);
   - impairing the other party’s likelihood of getting return performance (e.g., assigning a contract to a company that isn’t likely to pay the license fees).

   **P. Electronic Regulation of Access, Use and Repossession.**
1. **Online Access Contracts**: Upon material breach of an access contract, or if the agreement so provides, a party may discontinue all contractual rights of access of the party in breach [UCITA Section 814; Va. Code 59.1-508.14]. An “access contract means a contract to obtain by electronic means access to, or information from, an information processing system of another person...” UCITA Section 102(a)(1); Va. Code 59.1-501.2. The right to cut off access does not allow the licensor to retake prior transfers of software or information already delivered to the user, but merely to stop future access and performance [Id., Comment 3].

2. **Electronic Usage Restrictions.** UCITA Section 605, Va. Code 59.1-506.5, governs automatic passive restraints on software usage (sometimes known as software keys and hardware dongles). UCITA takes a tolerant view of automatic passive restraints and allows them if the:

- Agreement authorizes it, or
- Restraint prevents use which is inconsistent with the agreement, or
- Restraint prevents use after expiration of the stated duration of the agreement, or
- Restraint prevents use after a stated number of times specified in the agreement or
- Restraint is not a time-out or usage counter, but otherwise prevents use after contract termination and licensor gives reasonable advance warning before activation (this appears to be a catch-all provision and is the only one that requires advance notice).
**Don’t Block Access to User Data:** Automatic restraints may not be used if they affirmatively prevent the licensee from accessing its own information *through its own means (other than by continued use of the licensed program).* Note: Virginia’s version of UCITA deletes the preceding italicized wording. Va. Code Section 59.1-506.5 (as amended 2/19/01).

**Passive v. Active.** UCITA Section 605, Va. Code Section 59.1-506.5, authorizes passive restraints to enforce contract terms in the absence of a breach. An analog in the physical world would be a timing device that limits a Laundromat dryer to 30 minutes use if only a 30 minute duration was purchased [Offic. Comment 2].

**Example:** if licensee purchases a 30 concurrent user license, a passive restraint that prevents usage beyond the 30 user limit is permitted. However, a restraint triggered when the licensee tries to add a 31st user that affirmatively deletes the entire authorized copy and prevents any use (including the original 30 users) is considered overreaching and is not a permissible “passive” restraint.

**Example:** at the end of a one-year license, a passive restraint may cause the program to become inoperable and may erase the program. No prior notice is required for such automatic termination, because the license expired according to its own terms (there was no breach involved and the license authorizing any further
use expired). Remember not to erase the user’s data files.

Upgrade Replacements: When a vendor distributes new versions of an existing product, it is permissible for the new version to disable the old version as an incident to upgrading the user, if the agreement so provides for such replacements. Va. Code Section 59.1-506.5(e).

Example: The following license provisions identify the existence of the License File containing usage limiting code and then contractually links the scope of the license to the License File and the License Fee paid:

“License File. License File means a component of the Software that enables one or more other components of the Software and may also specify the location of the Designated Workstation and the Licensee. The License File also specifies certain limitations on the use of the enabled components of the Software and the purposes for, and extent to, which the enabled components of the Software may be used. Certain components of the Software may be licensed hereunder without a License File. For reference purposes, the components of the Software enabled by a particular License File correlate to Licensor part numbers as may be reflected on the License File or on one or more Order Forms.

Grant of License and Limitations. In consideration of the License Fee and subject to the terms, conditions and limitations set forth in this Agreement and the limitations set forth in the License File and any Order Form, Licensor grants to Licensee a nonexclusive license to use a single instance of the components of the Software specified in the License File and /or the Order Form for which the License Fee has been paid solely for Licensee's business operations on a single Designated Workstation at the location specified in the License File or Order Form. The Software is licensed to Licensee, not sold.”
Note: the use of passive restraints does not authorize the use of restraints to enforce remedies in the event of nonpayment or other breach of a contract. Passive restraints may be used to prevent use beyond expiration or non-renewal of a license, but may not be used to terminate usage of a license that is cancelled for breach, but has not otherwise naturally expired according to its terms, absent the breach (see Electronic Self-Help below).


(a) Physical Repossession. Unless software or licensed information has been commingled with other information, a licensor generally has a right to physical self-help under Section 815(b), Va. Code 59.1-508.15(b), where licensed software or content can be physically recovered without breach of the peace or damage to property.

Observation: This appears calculated to permit repossession of computer hardware with licensed software installed; stated conversely, a vendor is not precluded from physically repossessing hardware merely because it also contains the vendor’s software.

(b) Electronic Repossession. The rules for electronic self-help are much less accommodating to vendors than physical repossession. Absent an agreement, the use of electronic self-help upon breach of contract is prohibited.

- Mass Market Licenses: Electronic self-help is completely prohibited in mass-market transactions regardless of what the agreement provides. Mass-market transactions include: (a) consumer transactions,
and (b) end-user retail licenses (other than site licenses) of commercial off-the-shelf software or content publicly available on substantially similar terms, *but excluding* access contracts (e.g., web sites), and contracts to distribute or publicly perform/display copyrighted works.

- **Contractual Requirements:** Even when permitted by agreement, its use in a breach of contract situation (e.g., to “pull the plug” when the licensee doesn’t pay for custom software) is severely regulated by Section 816; Va. Code 59.1-508.16. The licensee must “manifest assent” separately to a contract provision authorizing electronic self-help (see Section 816 for complete details). It is not enough to manifest assent to the contract as a whole containing such a provision. Before resorting to electronic self-help, the licensor must give 15 days advance notice to the licensee’s designated representative in detail *(Note: Virginia’s UCITA requires 45 days, so check local law for possible variations on notice requirements; see Va. Code Section 59.1-508.16). Failure to comply with the notice requirements, among other exceptions, can expose the vendor to liability for consequential damages. Id. at subsection (e).*

**Bottom Line:** It’s a very risky to “pull the plug” on an operational system unless you carefully consider all of the rules under Va. Code 59.1-508.16. It is better to use an annual or other periodic license term with periodic software keys that must be issued to continue using the software (upon prior payment).
• **Public Policy Limitations:** the licensor may still not resort to electronic self-help if it has reason to know it will result in substantial injury or harm to the public health or safety or harm to the public interest substantially affecting third persons not involved in the dispute. The litany of exceptions is potentially so broad that it substantially restricts circumstances where electronic self-help may be used.

• **Invoking Public Policy Protections:** If a licensee receiving notice of an intention to exercise electronic self-help replies in good faith with its own notice to the licensor that any of the damages described above may occur, or if the licensor otherwise has reason to know of such potential damages, then the licensor will be **liable for any consequential damages** caused by its use of self-help. This is so even if the agreement excludes liability for consequential damages.

• It is therefore almost never advisable to use electronic self-help in a breach of contract situation, especially if the licensee can make a colorable claim that it will harm innocent third parties. The Courts will look to see whether you acted reasonably and in a balanced manner...
All Things in Moderation
Q. **“Entire Agreement” Clause.** This clause specifies what documents and communications form part of the Agreement. Product descriptions (by version number), pricing schedules and Statements of Work (SOW) for development projects are often included. Proposal documents are often excluded on the theory that the more specific exhibits cover those issues. Boilerplate contained in standard Purchase Orders and the like are also typically excluded. If a contract incorporates by reference external documents via an http: link, but sure you read the document and determine whether the vendor should have the ability to modify the document (e.g., online acceptable use policy or perhaps a privacy policy) without the customer’s express agreement.

Example: **“Entire Agreement.** This Agreement (together with any information from the Order Forms and License Files necessary to identify the Software that is the subject of this Agreement or further specific restrictions applicable to such Software) constitutes the complete and exclusive agreement between the parties and supersedes all prior or contemporaneous agreements or representations, written or oral, concerning the subject matter of this Agreement. This Agreement may not be modified or amended except in a writing signed by a duly authorized representative of each party. No other act, document, usage or custom shall be deemed to amend or modify this Agreement. It is expressly agreed that the terms of this Agreement and any Order Form issued by Licensor shall supersede the terms in any Purchase Order or other purchasing document submitted by Licensee; and the terms of any Licensee Purchase Order or other purchasing document are expressly rejected. Certain components of the Software may also be subject to a paper or electronic license agreement delivered by or on behalf of Licensor concurrently herewith, the terms of which shall be supplemental hereto to the extent not inconsistent herewith. If a copy of this Agreement in a language other than English is
included with the Software or Documentation, it is included for convenience and the English language version of this Agreement shall control.”

“This Agreement incorporates by reference the Privacy Policy contained at the following link: (http link here). The Licensor may from time to time revise the Privacy Policy and Licensee's continued use of the Software after Licensor posts written notice of the revised Privacy Policy shall constitute Licensee's acceptance of those terms.”

III. IMPORTANT ISSUES FOR INTERNATIONAL TRANSACTIONS

A. IP Protection. In some markets it can be extremely difficult, if not impossible, to monitor the manufacturing and distribution of your product when third party distributors are involved. In those cases, consider distributing only executable files manufactured by you and which are protected by license keys and/or dongles.

B. Governing Law & Forum Clauses. On the World Wide Web and in software transactions involving multiple states or countries, the choice of law and forum provisions may become the most important clause in the contract. This is because a favorable law/forum may inhibit the bringing of legal action in the first instance.

1. Governing Law. UCITA Section 109, Va. Code 59.1-501.9, allows the parties in their agreement to specify applicable law, even if it is in a neutral forum that has no relationship to the transaction. In cyberspace transactions where physical locations are often irrelevant or not knowable, “[p]arties may appropriately wish to select a neutral forum
because neither is familiar with the law of the other’s jurisdiction. In such a case, the chosen State’s law may have no relationship at all to the transaction." [Official Comment 2(a)]. The agreement to specify applicable law is limited by general concepts of unconscionability. Id.

Under the Model UCITA, if the agreement does not specify applicable law, then: (a) in B2B and B2C web-site or similar electronic access environments, UCITA assumes the transaction is governed by the licensor’s home state law (i.e., its primary place of business, rather than where the server is located), while (b) in transactions involving distribution of tangible software products in B2C context, the licensee’s home law is deemed to apply, and (c) in all other cases (e.g., distribution of tangible software products in B2B market), the law of the jurisdiction having the “most significant relationship” will apply. Under Virginia’s UCITA, “In the absence of an enforceable agreement on choice of law, the contract is governed by the law of Virginia.” Va. Code 59.1-501.9(b).

**Observation:** This preserves the traditional rule governing consumer mail order businesses, but spares “pure play” web-sites and online access providers (business or consumer) from having to comply with the law of every conceivable jurisdiction from which their sites could be accessed [Comment 3].

**Anti-UCITA Legislation Overriding Choice of Law:** At least four states (Iowa, N.C., W.Va. and VT) have passed legislation that overrides a choice of law provision that would select UCITA (Maryland or Virginia law). These statutes require that the
agreement be interpreted pursuant to their own laws if the party against whom enforcement is sought is a resident of, or has its principal place of business in, one of those four states.

2. **Governing Forum.** UCITA Section 110, Va. Code 59.1-501.10, allows the parties’ agreement to specify an exclusive forum for resolving disputes, but this selection must be specified in the agreement as exclusive and the selected forum must not be unreasonable or unjust. Under Va. Code 59.1-501.10(b), an exclusive forum selection clause in a mass-market transaction must also be conspicuous.

*Official Comments:* “Choice of forum agreements are generally enforceable...Agreed choices of forum are important in electronic commerce. Court decisions on jurisdiction in the Internet demonstrate the uncertainty about when merely doing business on the Internet exposes a party to jurisdiction in all States and all countries.” Comment 2-3, citing, Carnival Cruise Lines, Inc. v. Shute, 111 S.Ct. 1522 (1991)(upholding choice of forum clause in Cruise Ship ticket form). **Choice of a forum at a party’s location is ordinarily reasonable** [Comment 3].

**Examples:**

**Governing Law.** This Agreement shall be interpreted and enforced in accordance with the laws of the Commonwealth of Virginia, USA, without regard to choice of law principles.

**Governing Forum.** The parties irrevocably agree that: (a) any action arising out of or concerning the subject matter of this Agreement shall be initiated and maintained in a court of
competent jurisdiction located in Fairfax County, Virginia, USA (the “Forum”) and (b) each party irrevocably consents to exclusive personal jurisdiction in such Forum and waives any defense based on lack of personal jurisdiction, or improper forum or venue.

**Sticking Point?** If negotiations get bogged down over the choice of forum, consider a reciprocal provision that requires the party filing a claim to do so in the respondent’s home jurisdiction. Choice of law issues may sometimes be resolved by having U.S. law govern IP rights and contractual issues, while having the other party’s local law govern labor and employment law issues, if that is the location for on-site work.

3. **Arbitration Provisions.** You may want to consider adding an arbitration provision for resolving any disputes, or at least those disputes not requiring injunctive relief:

**Jurisdiction and Arbitration.** The parties irrevocably agree that, except for certain injunctive relief authorized under Section __, all disputes, claims or controversies arising out of or relating to this Agreement that are not resolved by the parties’ good faith attempt to negotiate a resolution shall be submitted to final and binding arbitration before JAMS/Endispute, or its successor, in Fairfax County, Virginia, USA, pursuant to the United States Arbitration Act, 9 U.S.C. Sec. 1 et seq. The arbitration will be conducted in accordance with the provisions of JAMS/Endispute’s Streamlined Arbitration Rules and Procedures in effect at the time of filing of the demand for arbitration. The parties will cooperate with JAMS/Endispute and each other in selecting a single arbitrator who shall be a former judge or justice with substantial experiences in resolving business disputes with particular experience in resolving disputes involving computer software. The costs of arbitration will be shared equally by the parties. The provisions of this Section may be enforced by any court of competent jurisdiction. The arbitrator shall not be
empowered to award damages in excess of, or inconsistent with, the liability limitations contained in this Agreement; however, the prevailing party shall be entitled to an award of all costs, fees and expenses, including expert witness fees and attorney’s fees, to be paid by the party against whom enforcement is ordered.

4. **Export Restrictions.** United States law regulates the transfer of technology to certain countries and persons. The following clause can help you comply with the law:

> “Export Restrictions. Licensee agrees to comply fully with all relevant export laws and regulations of the United States (the “Export Laws”) to assure that neither the Software nor any direct product thereof are (I) exported, directly or indirectly, in violation of Export Laws; or (ii) are intended to be used for any purposes prohibited by the Export Laws. Without limiting the foregoing Licensee will not export or re-export the Software: (i) to any country to which the U.S. has embargoed or restricted the export of goods or services, which currently include, but are not limited to Cuba, Iran, Iraq, Libya, North Korea, Sudan and Syria, or to any national of any such country, wherever located, who intends to transmit or transport the Software back to such country; (ii) to any end user who Licensee knows or has reason to know will utilize the Software in the design, development or production of nuclear, chemical or biological weapons; or (iii) to any end-user who has been prohibited from participating in U.S. export transactions by any federal agency of the U.S. government.”

5. **Foreign Corrupt Practices Act.** U.S. law prohibits the direct or indirect payment of money or anything of value to foreign officials, political parties or candidates for office for the purpose of influencing any official act or inducing an omission in violation of the official’s lawful duty, when made to obtain, retain or direct business to any person. International licensing agreements typically include a warranty and indemnity provision regarding each party’s compliance with the FCPA.
“The beatings will stop when morale improves.”
PART TWO: SOFTWARE SUPPORT AGREEMENTS

Ongoing support services are central to a software vendor's ability to deliver total customer solutions. Without support, the vendor limits itself to marketing products on a hit-and-run basis. By providing effective support, the vendor can become a valued long-term strategic partner to help the customer run its business "better, faster, cheaper" and more profitably. Market research shows that customers overwhelmingly want vendors to help solve problems, not push products.

As the Millennium Bug illustrated, some software applications written in the 1970s are still being used today. After only five years, a typical corporate user will spend more on software support than on the original license fee.\(^6\)

Because the useful life of software may continue for years or even decades, it is important for customers to address software support issues at the inception of the licensing relationship. This is especially true for users of mission-critical enterprise level applications.

Failure to address software support issues in advance in a written contract can lead to some costly “lessons learned” for a customer, both from the technical performance perspective and from a budgetary standpoint. These issues can quickly become political and “legal” in nature.

\(^6\) The cost of support for enterprise class software is about 15-20 percent of the original one-time license fee, per year. This does not include consulting fees to implement the software or to port customizations to new releases of base software released under a support policy.
The Board of Education
A. Software Support Agreements Under UCITA.

The Uniform Computer Information Transactions Act ("UCITA") has been enacted in Virginia (effective July 1, 2001, as amended) and Maryland (effective October 1, 2000). It was introduced, but not passed, in Arizona, the District of Columbia, Illinois, Maine, New Hampshire, New Jersey, Oregon and Texas.

UCITA Section 612 (Va. Code 59.1-506.12) deals with "Correction and Support Contracts," commonly known as software support and maintenance agreements. These add-on agreements should be distinguished from remedial work needed to correct a breach of warranty on the original software deliverable, and software repairs under a limited remedy to "repair or replace" defects.

UCITA also distinguishes between maintenance contracts to provide bug fixes for existing software already delivered, and a contract to provide updates and new versions to software in the future.

The progression of these themes may be depicted as follows:

**Considered Warranty Work:**

- Corrective work to cure a breach on original software delivery;
- Repair and replacement work under a limited remedy on original software;
**Considered Support Work:**

- Add-on contract to provide bug fixes to original delivered software;
- Add-on contract to provide updates or new versions adding new functionality.

**No Implied Contractual Right to Improvements:** UCITA Section 307 (Va. Code 59.1-503.7) makes clear that, unless the licensing contract specifies otherwise: "[a] party is not entitled to any rights in new versions of, or improvements or modifications to, information made by the other party." This means that the right to receive a specific upgrade (e.g., Version 4.0) and the right to receive unspecified future upgrades "when and if available" must be explicitly included in the contract, since UCITA will not imply these rights.

**Observation:** Unless the contract provides otherwise, a maintenance agreement to provide bug fixes to the original deliverable:

- Does not imply a right to receive updates and new versions adding functionality;
- Does not imply anything more than a "reasonable" response time, and
- Does not imply that the remedial efforts will necessarily "correct performance problems" [UCITA Section 612; Va. Code 59.1-506.12].

**No Implied Right to Alpha or Beta Versions:** A contractual right to receive updates means updates the vendor makes "generally available" to users (i.e., a "GA Release") [UCITA
Section 307; Va. Code 59.1-503.7]. This means a licensee entitled to receive updates or new versions under a support plan is not automatically entitled to participate in limited release alpha or beta versions of the software. Licensors should make this explicit to be safe:

Example: "Support' means . . . generally available Releases and related Documentation for the Software licensed to Customer if and when such items are generally released. Support does not include pre-production releases of Software, such as alpha or beta releases."

Beta Versions Do Not Create Express Warranties. UCITA Section 402 ("Express Warranties") indicates that any description, sample, model or demonstration of a final product, which is made part of the basis of the bargain can create an express warranty. Va. Code 59.1-504.2. However, the authors of UCITA recognize that beta versions do not create warranties because they are ordinarily understood not to be products in "final" form [Section 402, Comment 5].

No Implied Right to Training: Finally, there is no implied right to receive training or instructional support services from the vendor [UCITA Section 612(b); Va. Code 59.1-506.12(b)].

For these reasons, it is important for customers subject to UCITA (and probably those in other jurisdictions that may be looking to UCITA for guidance) to spell out fairly precisely in the agreement the type of support they expect. Specifically, the parties should indicate in the contract whether there is a right to receive:
• Bug fixes to the originally delivered software (4.x to 4.y)
• Specific functional upgrades to be released in the future (x.0 to y.0)
• Future unspecified upgrades "if and when available" (basic support)
• Alpha or Beta test versions (so-called Limited Availability releases)
• Training or instructional support services

**Watch Post-Signing Reassurances.** Licensors need to be careful about giving customers post-contract reassurances concerning updates and support services that may be at odds with the agreement and their pricing model. UCITA commentary states: "If language is used after the closing of the deal (as when the licensee on taking delivery asks for and receives an additional assurance), the assurance may become a modification of the contract. An agreed modification requires no consideration to be binding." [UCITA Section 303(a); Section 402, Comment 2; Va. Code 59.1-503.3].

**Code Changes Can Zap Warranties.** If the licensee modifies a software program, this will invalidate all express and implied performance-based warranties regarding the modified version. A modification occurs if the licensee adds code, modifies code or deletes code from the program. A modification does not occur if the licensee simply uses existing program features to configure software or make normal use a Toolkit to assemble programs from pre-existing components. No invalidation occurs if the parties are engaged in joint development and one party modifies the other's program as part of its authorized scope of work [UCITA Section 407; Comments 1-2; Va. Code 59.1-504.7].
B. Understanding Different Types of Software Support Transactions.

Understanding the issues raised by software support agreements requires one first to determine whether the transaction involves:

- Commercial off-the-shelf software;
- Core software that requires customization before it can be used by a licensee, or
- Custom software developed from scratch.

It also helps to understand whether support for custom or customized programs is being provided by the original licensor or by an independent consultant.

1. Packaged Off-the-Shelf Software. Direct licensing of off-the-shelf software products by the licensor to an end-user licensee is the simplest business model for support. For these purposes, "off-the-shelf" means the product is marketed as a stock item and can be installed and used with little or no customization. See SOP 97-2's definition.

For these transactions, licensors will generally want to "commoditize" their licensing and support practices, while shifting out-of-scope requests into its consulting practice. Standardization reduces transaction costs and produces more predictable demands on a customer support group. This makes the licensor's business model more scalable and profitable.

A licensor achieves this through the use of standardized agreements. The time and expense of negotiating agreements
can be further minimized by using click-wrap or shrink-wrap licensing mechanisms. Likewise, the licensor may use online registration for support as a way to "commoditize" its support offerings and minimize negotiation of terms. With more customer support implemented through self-help web sites and electronic communications, the licensor is better able to serve customers at all hours while regulating demands on live support resources.

Support for off-the-shelf software is usually priced as a percentage of the original up front license fee for the software. An annual support fee of 15-20 percent of the then current "list price" for the software is not uncommon. The licensee may want to include a price protection clause:

**Price Protection Clause:** "For a period of three (3) years from the Effective Date, the increase of the Support Fee for current Software programs for each subsequent Support Term shall not be more than five (5) percent plus the Consumer Price Index reported for the immediately preceding Support Term."

Some channel reseller agreements allocate responsibility for first line customer support (a.k.a., "first point of contact" or

---

7 Product support web sites are likely to involve an online "access contract" or "a contract to obtain by electronic means access to, or information from, an information processing system of another person...." [UCITA Section 102(a)(1); Va. Code 59.1-501.2(a)(1)]. Upon material breach of an online access contract, or if the agreement so provides, a party may discontinue all contractual rights of access of the party in breach [UCITA Section 814; Va. Code 59.1-508.14]. The right to cut off access does not allow the licensor to retake prior transfers of software or information already delivered to the user, but merely to stop future access and performance [Id., Comment 3].
"FPOC") to the reseller or to a third party customer service organization. FPOC typically involves fielding and answering simple support questions and forwarding unresolved issues to second line technical personnel. This removes the licensor's technical support people from the front lines of customer relations and allows them to focus more on technical level problem resolution.

The agreement should distinguish between responsibility for first line support and higher levels of support, especially when third party software is being distributed in object code form:

**Example:** "If ABC serves as the initial point of contact on Support services with respect to Third Party Software, then ABC will perform such services in a reasonable manner but without warranting the outcome, it being acknowledged that ABC may not have control over the resolution of issues for Third Party Software."

2. **Packaged Core Software with Customizations.**

Quite often, large enterprise level software programs require extensive customization and implementation services before a licensee can obtain beneficial use. For some complex software, customization and implementation services may cost 10-15 times more than the underlying license fee for the core software.

(a) **Core Software or Customized Software?** When software is customized for a specific licensee, it is important to distinguish between support for the unmodified base product and support for customizations created specifically for that
licensee. Quite often, when updates or new versions of the base product are released, they are not backward compatible with the customizations.

Example: "The Support Fee does not include services requested as a result of, or with respect to, causes which Licensor cannot reproduce on unmodified versions of the Software. If and when available, these services will be billed to Customer at Licensor's then current rates. Except as expressly stated in a separate Consulting Agreement between the parties, Customer is responsible for the migration of data and any customized versions of Software to new standard releases or versions of the Software issued as Support."

This means updates to the base product are useless unless the licensee is also willing to pay a consultant to port the customizations to the new base product. A licensee contemplating the cost of support for customized core software should therefore budget not only the cost of receiving periodic updates under the annual maintenance plan from the licensor (typically costing about 20 percent of the original license fee per year), but also the cost and time required to port any required customizations to the new base program.

(b) Customizations by Licensor or Consultant? There are also important control issues at stake when customizations are provided by an independent consulting organization rather than the licensor or the licensee. Many licensors maintain consulting alliance programs in which customizations are performed by independent consultants that have a technical alliance relationship with the licensor. An alliance consultant
may have access to source code, educational resources and technical support from the licensor's development team.

Alliance members are generally not "partners" with licensors in a legal sense. This means the licensor is probably not responsible for the consultant's performance, although problems with implementation often result in finger-pointing between the licensor, the consultant and the licensee's technical staff. It is therefore important for licensees to understand how these consulting alliances generally work at the contractual level.

For example, a consulting alliance member may have the right to modify the source code or at least create custom interfaces between the licensed program and other software, such as the licensee's legacy systems or third party products. The alliance agreement with the licensor often prohibits the consultant from delivering the customized source code to the licensee. Depending on the alliance agreement, the consultant may own its customizations and may (or may not) be required to deliver reference copies of the customizations to the licensor.

With only machine-readable object code in hand, the licensee will be in a position of long term reliance on the consultant's technical availability and willingness to provide support at reasonable prices. If the consultant retains control over the customizations, but is unable to provide adequate support to the licensee, this could harm both the licensee and the licensor's reputation in the marketplace. It's therefore

---

8 The "source code" is the human-readable version of the software that is needed to modify and perform ongoing maintenance and support work on the software.
important to understand the relationship between the licensor and any consultant in order to evaluate potential issues in long-term support contracts.

**Query:** If the consultant fails to cooperate or goes out of business, what options does licensee have to obtain support elsewhere?

From the licensee's perspective, it would be ideal for the licensee to try to obtain ownership of the customizations (which would include source code). This would reduce the licensee's dependence on the consultant.

Obtaining ownership may also help the licensee defend its competitive position if the licensee is worried that customizations it paid for could be re-used by the consultant on projects for "free riding" competitors of the licensee. A licensee may sometimes be able to secure ownership, especially if it is a large account with an IT department capable of implementing complex software and if disclosure of trade secrets in the source code by licensee is not a serious risk.

These ownership rights are usually difficult to secure in many cases, however. The licensor will generally want to retain control over its IP rights, which would include customizations and other derivative works. This is especially true if the customizations might be used as a framework or template for an entire industry segment. The consultant also will want to retain ownership of customizations to lock the customer into a long-term sole-source support arrangement and to re-use the customizations on other projects.
From the licensor's perspective, the ownership issue is often addressed in either the Alliance Agreement or any separate Source Code License between the licensor and the consulting alliance member. Thus, a licensee should explore whether the licensor or the consultant owns any customizations under their agreement. Otherwise, if licensee attempts to negotiate ownership of customizations with the consultant, it may be barking up the wrong tree.

As an alternative to ownership, a licensee could attempt to obtain a source code license for any customizations specific to that licensee. However, the licensor may block this effort because of concern over its trade secrets in the source code.

Another way for the licensee to reduce dependence on the consultant is for the licensee to make sure the licensor either retains ownership of customizations or receives the source code from the consultant and the right to modify and support the customizations. This may allow the licensor's consulting shop to provide support if the consultant proves uncooperative or too expensive.

**Source Code Escrows:** If none of these alternatives succeeds, the licensee may require that the source code for customizations be deposited into a "Source Code Escrow" account, with the code released if the consultant goes bankrupt or fails to provide ongoing support services.

The licensee should also determine whether the licensor has its own consulting group that can develop customizations. Licensor consulting shops will often have a very close working relationship with the licensor's development team and be more familiar with product technical capabilities. Although in-house
shops are sometimes perceived as costing more, their greater efficiency may more than offset the nominally higher rates. The elimination of potential finger-pointing between licensors and consultants may also justify using licensor consulting shops.

3. **Support for Custom Developed Software.**

Support for custom software is typically the final stage in a long process of designing, developing, testing, implementing and supporting a software system. It is important to understand the development cycle, since later stage support problems may simply reflect design flaws, coding errors or management problems at earlier stages of the project.

(a) **Project Staffing Issues.** It is important for both parties' technical people to get to know each other prior to awarding the contract. Sometimes, bringing in an outside developer intrudes on the turf of in-house technical people or other contractors who can undermine a project's success. The customer will want to review the resumes of contractor personnel to be assigned to the project and to assess their technical ability and experience with similar systems. Each party should specify a Project Coordinator and the key technical people. The agreement should include a non-solicitation clause to maintain workforce stability and should provide some level of reassurance regarding staffing stability and commitments, replacement/approval procedures, adherence to workplace safety, security and substance abuse policies.

**Learning Curves.** Having to replace project managers or key technical personnel during a project can have a
devastating effect on progress, quality of work and subsequent ability to diagnose and correct defects during support. Developers who quit may have gotten in over their heads technically, so their departure may signal problems with the project. Replacing key staff may entail a steep learning curve for the new people before they become productive. If the contactor removes staff without customer's consent, the contract may require a transition instructional period and deny or limit customer's obligation to pay for the orientation time. The customer may also restrict reassignment of contractor personnel to projects with key competitors for a specified period of time.

(b) How Good Software Design Affects Support. Many support problems can be traced back to the parties' failure to allocate sufficient time and resources to identify the customer's requirements and prepare functional and detailed technical designs (including interfaces between all systems) prior to actual coding.

A robust design phase allows you to unhitch design from development and is the foundation for successful development and debugging phases. Having a good design and clean, documented interfaces between software programs also helps demarcate lines of responsibility between any co-development teams and reduces finger-pointing. Good design can be imperative for support personnel later attempting to locate the cause of a defect and correct it.

**Structuring Tip:** Consider breaking the project into separate phases: requirements analysis, functional design, technical description, coding & testing,
implementation, training, support. Each phase must be completed, reviewed, accepted and paid for before the next phase commences. This discipline can be enforced by using a Basic Ordering Agreement (BOA) under which individual Task Orders are issued for each project phase. The contractor's need to prove itself on each Task Order before another one is issued imposes discipline and tends to keep contractors on their good behavior. The Task Order approach may also help the contractor if the agreement compartmentalizes contractor's liability to the amount paid under each Task Order.

(c) How Testing and Acceptance Effect Support. It is important not to put a software program into productive use until it has been vigorously tested and accepted. Licensees should try to resist efforts to characterize bugs identified during acceptance testing as something that will be "taken care of as a support issue." It is important to have a written testing and acceptance procedure to establish a clear line between warranty work and follow-on support services.

Acceptance testing should be conducted pursuant to a written test plan that is derived from the design documents. Each unit of code should be individually tested for functionality and reliability. The system should then be tested to see whether the units function together as an integrated system and with other systems.

As noted, efficient debugging usually depends on how well the system was designed from the outset, because a good design allows developers to isolate, identify and fix bugs quickly (sometimes within a few minutes). It is exasperating and time consuming for support personnel to try to support a poorly
designed or documented system. This is why a robust design is so important to the development of a functional and reliable system that can be supported effectively at reasonable cost.

There are two basic approaches to drafting an acceptance clause:

**Contractor version:** "The Software shall be deemed accepted if Customer does not notify Contractor of any material defects during the Test Period or, if Customer makes beneficial use of the Software, it shall conclusively be deemed accepted and any outstanding punch list items shall be covered to the extent provided under Section __ ("Warranties")."

**Customer version:** "The Software shall be accepted as conforming to the requirements of this Agreement only when Customer issues its written certification of acceptance. If Contractor fails within thirty (30) calendar days to correct punch list items or the Software otherwise fails to comply with the requirements of the Test Plan within the designated time, Customer may: (a) reject the Software in its entirety and recover amounts paid hereunder; (b) issue a "partial acceptance" of the Software, with an equitable adjustment in the price to account for such deficiency; (c) conditionally accept the Software, while reserving its right to revoke acceptance if timely correction is not forthcoming, or (d) pursue whatever other remedies are available under this Agreement."

**(d) How Confidentiality Restrictions Affect Support.** Human readable source code, which is required to fix bugs and
support software, constitutes a closely guarded trade secret. Even if the licensee obtains a source code license to provide its own support, a confidentiality clause may restrict access to the code to licensee's own employees. A licensee that does not have a robust IT department may want to loosen that restriction so its outside consultants can also provide needed support. The licensor will impose procedural controls giving it the right to approve or disapprove those consultants and a requirement that any consultant obtaining access to source code first execute a confidentiality agreement in a form supplied by the licensor.

C. **Other Key Software Support Issues.**

1. **What is an "Error" in the Software?** It is important to understand the definition of an "error" that gives rise to the duty to provide corrective service:

   **Licensee Version:** "Error" means any failure of the Software to perform its intended function as described in its related documentation or any significant inaccuracy in its related documentation.

   **Licenser Version:** "Error" means any reproducible failure of the standard, unmodified Software to perform substantially according to its related documentation. For these purposes, "documentation" means the official online help files or written instruction manuals supplied by Licensor regarding the use of the Software. Documentation does not include any FAQs, specifications for any custom development or integration services, any sales or marketing literature or any documentation for Third Party Software.

   91
2. **Know the Types of Software Releases.**

The following example (below) may be helpful for licensors charging an annual support fee that includes all future releases of the Software. If the contract includes Maintenance Releases in the annual support price, but charges extra for more substantive upgrades to the software adding new functionality (Minor and Major Releases), then the parties may want to negotiate over the bracketed phrase, which could give one party greater control over that issue:

"**Maintenance Release**" means a set of the Software [reasonably] [designated by Licensor] containing bug-fixes to existing functionality.

"**Minor Release**" means a set of the Software [reasonably] [designated by Licensor as] containing some new Software functionality and bug-fixes.

"**Major Release**" means a set of the Software [reasonably] [designated by Licensor as] containing substantial restructuring and major new functionality to the Software.

Another classification approach employs a more superficial naming convention within Licensor's control. Note that none of the definitions specify whether functional changes are included:

"**Software Release**" means a release of Software that is designated by Licensor in its sole discretion by a change in the digit(s) to the left of the decimal point in the Software version number [(x).x.x].
"Version Release" means a release of a Software that is designated by Licensor in its sole discretion by a change in the tenths digit in the Software version number [x.(x).x].

"Update" means a release of Software that is designated by Licensor in its sole discretion by a change in the digit(s) to the right of the tenths digit in the Software version number [x.x.(x)].

"Upgrade" means Updates, Version Releases, or Software Releases that Licensor makes generally commercially available.

3. **Understand the Different Levels of Support.**

It is important to understand that software support responsibilities, like a layered cake, can be split off and assigned to different divisions or companies. This ability to parse out support obligations can be especially important in reseller relationships. The following example identifies different levels of support that can be allocated between an owner of software and a reseller or an independent service organization. In this example, the reseller may also be licensing its own software to end-users.

**Example:** "Maintenance and Support shall consist of First Point of Contact, First Line Support, Second Line Support and Third Line Support, defined as follows:

**First Point of Contact** ("FPOC") means responding to End User support calls in the following manner: (i) the End User encounters a support issue and reports this issue to Reseller; (ii) the case is registered and logged into Reseller's support system; (iii) Reseller will assess the case and identify which Party's products are responsible for the support issue; (iv) if it is a Product support issue, Reseller will transfer the issue to Owner along with any necessary customer information and the case log will be updated with status assigned to Owner in the Reseller support system and the End User
will be informed of the transfer by Reseller; and (v) if it is not a Product support issue, Reseller will provide FLS.

**First Line Support** ("FLS") means the provision of assistance to end-users with regard to the supported products via telephone, fax, electronic mail or otherwise, including (i) identification of the source or cause of the support issue experienced by the end user (ii) whenever at its reach, directly provide a solution to the support issue received including fixed object and patch distribution, and, if applicable, (ii) escalation of the support issue to the Second Line Support.

**Second Line Support** ("SLS") the provision of assistance via telephone, fax, electronic mail or otherwise with respect to the supported products, including (i) receiving and processing support issues relating to the supported products as identified and escalated by the FLS, (ii) characterizing and analyzing support issues, (iii) registry administration - and whenever possible - resolution of calls, (iv) clarification of functions and features of the supported products, (v) clarification of the end user documentation for the supported products, and (vi) guidance in the operation of the supported products.

**Third Line Support** ("TLS") means the follow up activities related to support issues concerning the supported products as escalated by the SLS and which entail but are not limited to (i) further and detailed in depth analysis of the said support issues leading to a) well documented and accurate description of the support issue at stake, b) replication of the said support issue on a reasonably acceptable test system c) documented case resolution path, (ii) deployment of efforts till the case is resolved either by means of a workaround or the intervention of product engineering."

**4. Service Hours of Support.**

If the licensor's support group is located several time zones away from the licensee's business operations, then it is important to specify in the contract which local time applies to
any normal hours of support. For global operations, it may also be important to specify what language applies to support.

5. Classification of Support Calls.

Support calls or "Cases" are usually classified according to their level of severity, as measured by the impact of the reported problem on the licensee's business operations. It is important to know which party has the power to classify the support call.

The licensor will generally want the discretion to classify or at least verify the error, while the licensee will want either to classify the error or at least to apply an objective standard (this objective standard may need to be reflected in the definition of "Error"). Below is a fairly evenly weighted classification approach that attempts to balance each party's interests (bracketed language reflects some of the phrases that might be negotiated):

Example: "Licensor will [assign qualified, experienced technical staff and] make its [best] [reasonable] effort to correct Errors that Licensee identifies, classifies and reports to Licensor [and that Licensor substantiates]. Licensor may reclassify Errors if it [reasonably] believes that Licensee's classification is incorrect and can substantiate same to Licensee's [reasonable] satisfaction."

6. Some Typical Support Case Classifications:

"Case Category 10" means that Licensee's live system is at a halt and is unable to process data through the Software as a result of a catastrophic
event in the system database or Software, or a major application failure in a critical processing period.

"Case Category 20" means a problem in the Software which causes serious disruption of a major business function and which cannot be temporarily solved by a workaround.

"Case Category 30" means any of the following: (i) a non-critical problem in the Software where the Licensee is able to continue to run the system and/or application or a workaround is available; (ii) a reported problem in the Software that does not qualify as a Case Category 10 or 20.

"Case Category 40" means all questions and requests for information on the Use or implementation of the Software.

7. The Importance of Defining "Response Times."

Licensors will naturally be expected to respond to Support Cases according to their severity. Licensor will want to measure their "response time" from the time it takes Licensor to begin diagnosis and error correction activities. Licensees may attempt (usually unsuccessfully) to measure response time to when the bug is fixed:

"Case" means an incident related to the operation of the Software as logged within Licensor's support system.

"Response Time" means the elapsed time between the receipt of a Case and the target time within which Licensor begins Support as verified by a verbal or written confirmation to the Licensee. Support is available during Service Hours with the following Response Times: (i) Case Category 10: one (1) hour; (ii) Case Category 20: two (2) hours; (iii) Case Category 30: four (4) hours; and (iv) Case Category 40: eight (8) hours.

8. Using Escalation Procedures to Attract Needed Attention.
Customers of large software installations or mission critical systems sometimes negotiate escalation procedures. These can help elevate a stalled support problem to a higher level within the vendor organization until it is resolved. Although these procedures in themselves do not guarantee the outcome, they do ensure that individuals at each level feel political pressure from above to perform.

"Escalation Procedure. If Licensee believes that Licensor is not satisfactorily performing its support obligations as specified in this Agreement, Licensee shall have the right but not the obligation to contact the following personnel listed below on a time frame to be determined by Licensee in its sole discretion ("Escalation Contacts"): 

- Level 1 Contact: Second Line Support personnel
- Level 2 Contact: Third Line Support personnel
- Level 3 Contact: Licensor Project Manager
- Level 4 Contact: Director of Development
- Level 5 Contact: Vice President of Development
- Level 6 Contact: President of Licensor
- Level 7 Contact: President of Licensor’s Parent Company

If Customer is unable to contact a particular person listed above, Customer may contact other personnel on such list immediately at its sole discretion. [The Escalation Contacts will make best efforts to take all steps (including without limitation assigning additional personnel or labor, expertise, equipment, materials or other resources, as required) that are necessary to have the reported problem addressed as set forth herein.]"
9. **Certain Licensee Responsibilities.** Most software licensors will include provisions in their support policy to ensure that it does not take responsibility for Support Cases that arise from the licensee's own failure to observe good operational practices. The sample below reflects "lessons learned" by one licensor:

"**Responsibilities of Licensee.** Licensor’s provision of Support to Licensee is subject to the following:

(a) Licensee shall provide Licensor with necessary access to Licensee’s personnel and equipment during Service Hours. This access includes the ability to dial-in to the equipment on which the Software is operating and may also include the ability to obtain the same access to the equipment as those of Licensee’s employees having the highest privilege or clearance level.

(b) Licensee shall adopt and utilize all updates, releases and enhancements offered to Licensee previously and adhere to the Support service policy statements that might be released from time to time.

(c) Licensee shall provide supervision, control and management of the Use of the Software. In addition, Licensee shall implement procedures for the protection of information and the implementation of backup facilities in the event of errors or malfunction of the Software or equipment.

(d) Licensee shall document and promptly report all detected errors or malfunctions of the Software to Licensor. Licensee shall take all steps necessary to carry out procedures for the rectification of errors or malfunctions within a reasonable time after such procedures have been received from Licensor.

(e) Licensee shall maintain a current backup copy of all programs and data and any programs or data that may be affected by the Software or data provided hereunder.
(f) Licensee shall properly train its personnel in the use and application of the Software.

(g) Licensee shall obtain access, to the World Wide Web at its expense, in order to access Licensor’s Global Support Web-site.

(h) Except to the extent expressly governed by a separate consulting agreement with Licensor, Licensee or its authorized designee is responsible for the migration of data and any customized versions of Software to new standard releases or versions of the Software issued as Support. Licensee will observe all Release Notes and product communications issued by Licensor.

(i) If Licensor serves as the first point of contact on Support services with respect to Third Party Software, then Licensor will perform such services in a reasonable manner but without warranting the outcome, it being acknowledged that Licensor may not have control over the resolution of issues for Third Party Software."

10. **Service Level Credits.** Service Level Credits are an administrative mechanism and compensation method (think of liquidated damages) for relatively minor performance problems. Service Levels could be focused on response times for support calls (different response times depending on the magnitude of the glitch) or system availability metrics for online applications (e.g., 99.5% uptime commitment, except for schedule maintenance or force majeure).

**Note:** be careful how “force majeure” is defined, as a broad definition can effectively gut an SLA provision. The vendor will try to include a broad array of possible problems, such as disruption of third party telecommunications, third party software issues (which could mean operating systems, database software and a
myriad of other layers), labor disputes (key workers leaving?) and other matters.

If the Licensor fails to meet the Service Level commitments, then customer usually receives a "Service Level Credit" as a limited remedy. Usually, the service level credits are applied to the amount otherwise due from the Licensee (e.g., 5%, 10%, 20% of the monthly invoice amount). Sometimes Service Level Credits are subject to a maximum credit limit for any one Support Case or for any specific month or other specified period.

If the Service Levels are not met for, say, any 2 consecutive months, or 3 months in any 5 month period, the agreement might provide a right for the customer to terminate for convenience and the right to recover a refund of any pre-paid fees (pro-rated from the termination date through the end of that Term). This approach avoids having to declare the vendor in breach, and gives the customer an easy exit from the relationship.

11. Minimum Period of Support. It is usually helpful for both parties if the Licensor specifies the minimum period of support for products that are discontinued. A common approach is to offer ongoing support for one or two years after product discontinuation:

   **Example:** "Licensor shall continue to offer Support for Software for a period of two (2) years after discontinuing a version of Software from Licensor's price list."

Both parties should understand how financial accounting rules governing recognition of revenue for software-related transactions affect the structure and timing of software license and support agreements. In some cases, these revenue recognition rules strongly shape the licensor's business model and the structure of its agreements.

**Practice Tip:** Paying attention to the mechanics of booking revenue can help licensees obtain period-end discounts from software companies trying to meet revenue targets. It can also help licensor attorneys understand the importance of maintaining good contracting, shipping and account receivable collection practices.

1. **License Fee Revenue Recognition.** A licensor may generally recognize a one-time software license fee as revenue when the contract has been signed, the product has been shipped, the fee is fixed or determinable and the invoice has been issued, provided collection of the money is probable within a normal payment cycle (e.g., net 30-60 days).

- **Close the Contract:** to help recognize license revenue and have it "stick" on audit, many licensors prefer to use standard agreements to evidence the transaction. When signed license agreements are generally used by a particular licensor, they must be signed by both parties to be effective (even if a purchase order has been obtained). This means licensors should have an officer available to execute agreements prior to the close of
each accounting period. If the licensor does not usually obtain signed license agreements (e.g., in a click-wrap licensing scenario), then the transaction can be evidenced by a non-cancelable purchase order from the licensee or, in the case of online sales, the transaction can be evidenced by an online acceptance of the order. Remember: failure to close the sale contractually will delay revenue recognition even if the product is shipped and payment is received.

- **Set the Price:** to recognize the revenue, the price must be fixed or determinable at the outset. Granting extended payment terms beyond 3 to 6 months can defeat this requirement, since the technological obsolescence of software over relatively short periods can increase the chance of having to grant concessions. Extending payments on 10+ percent of the fee beyond 12 months can lead to a presumption that the whole fee is not fixed and determinable at the outset and can delay revenue recognition until payments are received. A history of granting concessions to collect extended payments likewise can defeat immediate recognition. Ideally, the licensor will want the fee paid within 30-60 days of delivery to recognize the license fee up front.

- **A/R Factoring:** selling the account receivable to a third party financial institution, even on a non-recourse basis (in which payment risk is shifted to the bank), generally does not help the licensor recognize revenue on an extended payment license, since the "fixed or determinable" requirement is determined at the outset of the license transaction. While credit risk is
eliminated, the risk of technological obsolescence could still require future concessions.

- **Cancellation Privileges:** substantive cancellation privileges, such as the contractual requirement for board approval to an agreement, can delay revenue recognition. However, short term return policies (e.g., "30 day money back guarantees") generally do not interfere with revenue recognition of the price (minus a reasonable reserve for returns).

- **Deliver the Product:** the software product must be delivered to the licensee before the licensor can recognize the license fee revenue. If shipment is by common carrier, shipping the product "FOB Shipment Point" satisfies this requirement at the time the product is put into the hands of the carrier. "FOB Destination Point," however, can delay the delivery time until the product is actually received by the licensee. It is important for licensors to document the time of shipment by keeping shipping receipts on hand. For software delivered electronically, the delivery time occurs when the software can be downloaded or when license keys to a master copy are issued. So if it's too late for a Fedex shipment, consider delivering the software electronically.

- **License Keys:** the routine use of temporary license keys to enforce payment terms will not delay revenue recognition, provided the basic criteria are otherwise met. However, if temporary keys are not routinely use, their presence in a transaction may indicate there's a serious payment risk on that
particular transaction and delay recognition of the revenue.

• **Acceptance Procedures:** when software deliveries are subject to formal acceptance procedures by the licensee, this can delay the revenue recognition if acceptance is a substantive step, rather than a perfunctory matter. Postponing the license fee payment until acceptance indicates that the acceptance event is substantive and can delay revenue recognition. Conversely, including a "deemed acceptance" clause upon the passage of time suggests it is perfunctory. Note: a temporary "try before you buy" arrangement will probably delay recognition of the revenue until the trial period is over.

• **Remember to Get Paid:** a history of non-payment by a particular customer may defeat the "probability of payment" requirement for revenue recognition. Customers with a clean payment history can get better deals from software vendors because there is less qualitative concern over recognizing revenue on the sale. Likewise, a vendor with a poor track record of collecting accounts receivables at face value may have difficulty persuading its accountants that the next deal should be recognized when the recognition criteria have otherwise been met. Attorneys called in to help clean up old A/R on the books should realize that they may be helping the company improve its ability to recognize revenue on deals coming through the sales pipeline.

2. **Post-Contract Support Revenue Recognition.**
Support versus Specified Upgrade Rights: Post-Contract Support (commonly known as support and maintenance services) should be distinguished from a contractual right to receive a specified upgrade in the future that will add functionality or enhance performance of the software. For example, if an agreement licensing Version 3.0 grants licensee the right, at no additional cost, to receive Version 4.0 "when and if it becomes available," a portion of the revenue on the original license fee will need to be deferred until that upgrade is delivered. Note: licensor may be able to counteract this by including an "entire agreement" clause and a statement that all upgrades are at the "sole discretion" of licensor.

**UCITA Warning:** As noted earlier, licensors need to be careful about giving customers post-contract reassurances concerning updates and support services that may be at odds with the agreement and their pricing model. UCITA commentary states: "If language is used after the closing of the deal (as when the licensee on taking delivery asks for and receives an additional assurance), the assurance may become a modification of the contract. An agreed modification requires no consideration to be binding." [UCITA Section 303(a); Section 402, Comment 2].

**Post-Contract Support:** generally relates to unspecified upgrades that are to be provided "when and if available" and ongoing error correction services. For example, if the contract promises to deliver to the licensee all upgrades to a licensed product generally released during the next year "when and if available" (and none have actually been specified yet), then this is considered post-contract support rather than a right to a specified upgrade.
The value of support should generally be recognized ratably over the support period, using a straight-line method. In determining this value, the actual price designated for support may or may not reflect its true value for revenue recognition purposes. Accountants will often look at the renewal rate to be charged upon commencement of the second year as a better measure of how much of the total up front price should be allocated to support for year one.

**Coordinating Warranty & Support**: if a one year support period commences after completion of, say, a six month warranty period, then a fair portion of the total fee will be recognized over 18 months comprised of (a) an implied 6 month support term, followed by (b) an express support term of 12 months. The licensor will therefore have an incentive to move directly into product support upon inception of the license agreement.

**Watch the Gap Between License, Support & Hosting**: It is possible that software may be licensed from one vendor, supported by a different vendor and hosted online by a third vendor. This can create a gap in responsibility, where each vendor blames the other for performance problems. A slow system, for example, could be the fault of the software vendor (poor architecture or coding), the support vendor (inadequate knowledge of system design) or the hosting provider (insufficient allocation of memory or bandwidth) or the customer’s slow connection. It is therefore important to designate one vendor as primarily responsible for “system” performance and to
be responsible for coordinating the technical work of other vendors.

PART THREE: OPEN SOURCE SOFTWARE

I. What is Open Source Software?

Open Source is a development method for software that harnesses the power of distributed peer review and transparency of process. The promise of open source is better quality, higher reliability, more flexibility, lower cost, and an end to predatory vendor lock-in. [Quoted from the Open Source Initiative at http://opensource.org].

Despite its benefits, Open Source Software (OSS) is not risk-free. Depending most importantly on the specific OSS license that governs a particular work, the lawyer for a client company whose developers may be using OSS must be cognizant of the primary risk that creation and distribution of a derivative work of the OSS could under certain circumstances obligate the client to publish source materials for its proprietary programs that are based on or utilize the OSS.

Aside: Va. Code 59.1-504.10 indicates that there are no implied warranties of non-infringement, non-interference, or merchantability for “free” (open source) software for which the licensor does not intend to make a profit from its distribution and does not generally seek commercial gain for making, modifying or redistributing copies.
Because there are many varieties of OSS licenses, the lawyer must review the specific OSS license governing a particular OSS work in the context of the client’s intended use of the OSS. This involves a case-by-case analysis in many cases. Ideally, this should be done in the context of the client company’s Open Source Software Policy, where the issues are thought through in advance. The text that follows “shadows” the provisions typically found in such a Policy (a copy of the Policy draft is available for download).

II. Open Source Software Policies

The purpose of an Open Source Software Policy (the “Policy”) is to provide guidelines and procedures that client Company employees (typically in-house software developers) should follow before using, or allowing an outside software vendor to use, Open Source software in any (a) Company internal production systems or (b) any Company product or web service supplied to customers or made publicly available.

- Pre-approval is not required for Company staff to conduct an internal technical evaluation of OSS prior to actual development use.

- The approval process may be expedited for OSS licensed under certain “Permissive” (Academic) OSS licenses, as described below.

- For OSS works governed by “CopyLeft” or “viral” OSS licenses, as described below, permission will likely be denied unless it is certain the OSS would never be modified or combined with other software and redistributed (including via web applications).
Company employees who are considering the use of OSS must submit a Request for Approval to the OSS Review Board and obtain written approval prior to using, modifying, combining or linking OSS with other code or libraries, or distributing OSS derivative works as part of Company products or web site services.

III. Definitions to Know

One of the challenges of evaluating OSS licenses is the lack of uniformity in definitions and the legal concepts they express. It is essential to review the definitions contained in a particular OSS license in order to understand what obligation the user may have to publish otherwise proprietary code with which OSS may be technically associated in different ways:

Open Source Software (“OSS”) is software freely available in source code form for anyone to use, copy, modify and distribute without payment of a license fee or royalty. OSS is not public domain software. Instead, it is copyrighted software. Its use is governed by a particular OSS license selected by the original author, who has the right to control downstream uses of the work as a condition of granting the license. As each contributor makes improvements under certain kinds of OSS licenses they, too, must grant downstream users the same license rights to the latest derivative work.

Derivative Work is “a work based on” the OSS work that would require permission of the copyright owner (other than an exact copy). This would include “copy[ing] from or adapt[ing] all or part of the work.” A Derivative Work therefore includes (a) OSS code that is modified at the code level, or (b) OSS code that is translated or adapted into another work, or (c) the combined
work that results from incorporating OSS code (e.g., including snippets) into other code. Additional points to consider:

- Some OSS licenses define Derivative Work (or a “work based on” the original) at the file level, so that copying OSS code into another work only “taints” the other work within that same file. Other OSS licenses would taint the entire work, even including software whose only technical connection is that it statically or dynamically linked to the file containing OSS.

Distribute or convey generally means any kind of propagation of a work that enables other parties to make or receive copies. Generally, mere interaction of server software with users through a computer network is not a distribution or conveyance of a copy.

- Exception: The GNU Affero GPL provides that “if you modify the Program, your modified version must prominently offer all users interacting with it remotely through a computer network... an opportunity to receive the Corresponding Source....”

Source Code generally means “the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.” (Apache License v.2.0). OSS licenses that carry an obligation to publish “corresponding” source code for a derivative work may employ an expansive definition of Corresponding Source: “all the source code needed to generate, install and run the object code and to modify the work, including associated scripts, interface definition files, shared libraries and dynamically linked subprograms that the work is specifically designed to
require....” (GNU GPL v3.0) (emphasis added). Through this expansive definition, the GNU GPL v3.0 broadens the source code delivery obligation to encompass other software that may only link to or interact with it.

IV. Identifying the Governing Open Source License & Relevant Issues

Authors have discretion to choose from over seventy (70) “Open Source Initiative Approved” OSS licenses (overall, more than 1,400 varieties of licenses have been reported). The OSS files for a particular work will identify the exact OSS license that governs it. The 70 major OSS license documents are published at the Open Source Initiative website (http://opensource.org). Some authors publish OSS under more than one open source and/or commercial license to offer different development and support paths for their work.

A. OSS License Classification

OSS licenses fall within three general categories or classes:

• **Class A: Academic or “Permissive” Licenses**: OSS code governed by Academic Licenses (Class A) are usually deemed safe for Company use, modification and distribution with proprietary products and on standard commercial terms of Company’s choosing. Among the more popular Class A licenses are:
  • BSD License (v.2, 3);
  • MIT License;
  • Microsoft Public License (MS-PL).
Class A licenses allow a Company to decide how to license any derivative work of the OSS on an outbound basis of its choosing; that is, Company may license its customers executables only and it need not publish any source code to the derivative work of the OSS or any of its proprietary code. The Company does have notice and attribution responsibilities on distributed derivative works of OSS, as noted further below.

- **Class B: File-Based Licenses:** Under these licenses, OSS obligations to publish source code to distributed derivative works are imposed only at the File Level and do not taint other code that links to or interacts with the modified OSS file. Class B licenses include:
  
  - Apache License (v.2.0);
  - GNU Library or Lesser General Public License (“LGPL”);
  - Mozilla Public License (Version 2);
  - Common Development and Distribution License (Version 1);
  - Microsoft Reciprocal License (“MS-RL”).

The Apache License (v.2.0) accomplishes this by defining a derivative work to exclude programs that “remain separable from, or merely link (or bind by name) to the interfaces of, the [OSS Work].” The GNU LGPL permits distribution of a Combined Work that contains a separate Application that interfaces to an OSS Library. The licensee is obliged to publish source code only to the OSS Library (and not to the Combined Work containing the separate Application). The Mozilla PL (v2) simply limits the source code disclosure requirement to the
original Covered Software (the OSS) and any Modifications to it at the file level (code changes or copying OSS code into other software at the file level).

- **Class C: Copyleft Licenses (Prohibited at Company):** Under CopyLeft licenses, the use of OSS code to create a distributed derivative work not only requires publication of source code to the modified files, but to the entire “derivative work” or “combined work” which includes code with which OSS code is compiled and (according to the FSF) any other software to which OSS links, either statically or dynamically. OSS provided under a Class C license risks “virally tainting” other proprietary applications with obligations to disclose the source code to the proprietary or non-OSS programs. It could also conflict with third party licenses, including other open source licenses to other components with which it interoperates. Class C licenses include:

  - GNU General Public License (GPL, Version 2 and 3);
  - Eclipse Public License (Version 1)(possibly viral, depending on what the undefined term “derivative work” encompasses);

There is a debate whether the Class C “CopyLeft” licenses there cross-contaminate applications/plug-ins and Javascripts that are “interpreted” source code relied on by other applications (the outcome apparently depends on a fairly obscure technical analysis). Because of these risks, OSS under a Class C (CopyLeft or viral) license is not permitted at most companies unless it is certain the OSS would not be modified or combined with other software and redistributed (including via web apps).
Introducing “Class C” Open Source Software Into Your Code Base Can be a Ticking Time Bomb
The following discussion focuses on additional issues that would be relevant in considering Class A (Permissive) and B (File-Based) licenses.

**B. Distribution of Derivative Works**

If it is determined that the Company will create a “derivative work” (or some other defined work that invokes the OSS license obligations) then it becomes necessary to determine whether the resulting work will only be used internally by the Company, or instead, will be distributed or conveyed to third parties.

Distribution or conveyance of a copy of software is typically an obvious event, occurring either by physical delivery of software media or direct download of a copy. Software-as-a-Service or cloud-based remote applications accessed via the internet might (at least arguably) be considered “distributed” to the extent that software plug-ins and client-side software are downloaded to client workstations as part of using the service.

**Aside:** In Class C licenses (which are generally disfavored at the Company), it is noteworthy that “distribution” may include web or network access to pure server-side functionality under the GNU *Affero* General Public License clause.

It is the “distribution” or “conveyance of a copy” of a derivative work of the OSS that invokes the obligations to provide notice and attribution under Class A licenses, and in addition, to publish source code under Class B (at the file level).
C. **Notice and Attribution Responsibilities**

If the Company were to distribute a derivative work of OSS under a Class B (or C) OSS license, the governing OSS license will provide instructions on the notice and attribution requirements for that derivative work, along with directions for publishing the required source code. Attribution (recognition afforded the author of the derivative) is an important part of the reward structure of the OSS model.

D. **Software Warranty and Support Considerations**

When comparing OSS to commercial software, consider that much OSS is provided “as is” without warranties and indemnities typically found in commercial software licenses (e.g., warranties of non-infringement, performance, support, etc., are usually lacking). The general absence of IP indemnity rights in many OSS licenses poses a greater risk if infringing content has been contributed by an upstream coder. Regarding the absence of software support, in some cases, an independent software support organization may address this risk by offering commercial grade support for OSS for a stated fee.

E. **The OSS Project Sponsor**

OSS projects must have a Company sponsor. The sponsor is the individual who first requests to use the applicable OSS, or other individual that may be specified by the Review Board. The OSS sponsor will be responsible for:
• Managing any external support from the OSS community or support vendor, and for tracking and applying updates and security patches;
• Approving and registering modifications to OSS through Company’s source code control system;
• Notifying the Review Board when the OSS is no longer suitable or no longer used (end of life);
• Coordinating any approved OSS community interaction and distribution of source code with respect to the OSS.
• Ensuring that individual developers at the Company who make contributions to OSS projects approved by the Company receive proper attribution and recognition, as anticipated in the OSS license.

F. Record Keeping and Code Management

The Company source code control system should hold code, documentation, revisions and revision logs for all such OSS. When OSS components are used only in binary form, the license, source code and build tools needed to recreate that binary must, if available, also be archived in the Company source control system. These elements must also be archived for each new release of the applicable OSS.

Because many OSS applications are dependent upon older versions of libraries or other OSS components, an archive copy of all previously approved versions must remain in the Company source control system. OSS repositories within the Company source control system should be backed up periodically. When any approved OSS is no longer used or
needed, copies of the OSS should be removed from all company computers, except for archival copies.

Company software developers should search for OSS in the Company source control system before seeking to obtain OSS from an external source. OSS components in the Company source control system have already been evaluated and can be approved more quickly.

G. Compliance with OSS Licenses -- If OSS is approved for a specific project, the terms of the OSS license agreement must be followed. This may require compliance with notice and attribution (author recognition) requirements, distribution of source code but only for modified files (Class B licenses) and other terms of use. Since OSS license terms vary, the Legal Department should be contacted if there are any questions regarding a particular OSS license. Source code for anything beyond the immediately modified OSS file should never be released without legal department approval.

H. Employee Communication, Education and Training -- The Company may implement communication (and education and training if necessary) for development, IT, procurement, legal, and executives regarding the general use of OSS, how to minimize undue risks associated with use of OSS, the terms of this Policy and other pertinent topics.

I. Software Developed by Vendors -- Third party vendors engaged to develop software for Company should also provide representations, warranties, and indemnifications stating that the developed software does not contain any OSS, unless otherwise approved by the Company in the contract.
J. **Patent Considerations** -- Some OSS licenses contain patent licenses or patent non-assertion covenants. For organizations that hold patents and modify OSS into a work that is covered by their own patent, the patent provisions of the OSS license may be construed as granting patent rights on distributed OSS to downstream users on a royalty free basis. *See, e.g.*, the Apache license (v.2) and the GNU GPL v. 3. This is a risk that should be carefully considered by patent holders.

K. **Audits** -- Company may perform periodic audits to determine compliance with this Policy. These audits, which may include code scanning, will determine if this Policy is meeting business objectives and whether Company is conforming to the terms of this Policy.

L. **Violations** -- Any Company personnel found to have violated the terms of this Policy shall be subject to disciplinary action, including possible termination.

M. **Exceptions to Policy** -- Any exceptions or deviations from this Policy must be approved in writing, and in advance of any deviations, by either the Review Board or the Company General Counsel.

N. **Revisions to Policy** -- Company reserves the right to modify this Policy at any time, with or without notice. This Policy is confidential information of the Company.

O. **Roles and Responsibilities** – The policy should identify who is responsible for implementing this Policy, who “owns” document and related matters:

<table>
<thead>
<tr>
<th>Role</th>
<th>Responsibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>All Employees</td>
<td>All Directors and above</td>
</tr>
</tbody>
</table>
# Related Documents and References

Request for Approval to Use Open Source Software (OSS)

## Points of Contact

Questions or comments related to this Policy should be directed to _____ (___@company.com).

## Document Control

**Document Owner**
The Company ___________ (___@company.com)

**Document Location**
For the most recent approved version of this document, please see the Company Intranet. Consultants can view the policy at the non-employee portal.

## Audience Distribution List

When this Policy is updated the following audience must be advised through email that an updated copy is available on the Company intranet.

## V. Opportunities and Traps in Dealing with Open Source Software (OSS)

Clever software companies have realized that if you put OSS onto a cloud computing platform and put a pretty face (user interface) on it, it is indistinguishable from commercial software and, therefore, users can be charged commercial subscription prices for what amounts to free OSS software.

Vendors realize that most users do not bother to read the online Terms of Service or other license agreement, which
would contain the standard OSS public notices and alert them to the presence of free OSS.

Users are accustomed to paying commercial rates and are focused on the benefits received from using the software. That is the usual analysis when justifying a corporate budget line-item for software acquisition. But in the OSS world, focusing on “benefits received” is the wrong analytical model. Many OSS programs deliver tremendous benefit, but they are free for the taking. Why pay for it?

Instead, pricing should focus on what, exactly, does the vendor contribute that goes beyond the functionality of the OSS components?

The business model of shrouding free open-source software behind a pretty web interface and charging commercial rates is a clever slight-of-hand that off-loads development and support costs onto the open-source community while reaping windfall profits from unsuspecting customers who apply incorrect pricing models and assumptions.

**EQILOGUE:** Thirty years ago, I described “computer law” simply as “the work done by a lawyer who handles whatever problems land on his desk” often scrambling to help clients capture opportunities or manage risk. This often involved relatively “mundane and ancient areas of law that have been dusted off, uncorked and drained by the practical demands of the computer industry. In many cases, those demands have forced new concepts into old bottles that are simply too small to contain them…." The law has expanded rapidly in 30 years trying to keep pace with the computer industry. I wonder what it will look like 30 years from now?
“If I must spend time in this bottle, the first thing that I’d like to do, is kill all the lawyers.”