In today's technology-rich environment, in-house corporate counsel must necessarily handle a range of agreements that implicate intellectual property issues. Here are five key IP considerations to be aware of when retaining a contractor to develop software for your company.

1. Get Ownership Right

Companies can benefit from owning the IP rights in developed software; IP owners can use, commercialize, and modify the software free of the scope restrictions and termination risks often associated with license agreements. However, developers may have legitimate reasons to retain IP ownership; for example, they have invested knowledge capital in the project and may be able to reuse code or build on it for other client projects. Before insisting that IP ownership is a “must,” consider whether the software will be competitively sensitive (i.e., will it matter if the developer creates something similar for others?) and whether your company is likely to license it to others or otherwise commercialize it. If not, factors such as vendor bargaining power, cost savings, or even earning developer goodwill may be sufficient reason to allow a developer to retain IP ownership and grant a broad perpetual license to your company.

Beware of joint ownership: Jointly owning software IP with the developer can seem like an efficient way to sidestep difficult negotiations. Unfortunately, parties rarely consider the full implications of this choice. Joint ownership rules vary, not only by type of IP (e.g., patents, copyrights, or trade secrets), but by country. For example, under U.S. law each joint copyright owner may commercialize a copyrighted work without the other joint owners’ consent, but must account for licensing royalties received and may not destroy the value of the work. This is different from English copyright law (under which joint copyright owners cannot exploit their rights without the other joint owners’ consent), as well as U.S. patent law (under which joint owners have no duty to account to each other for licensing royalties). Parties can modify these default rules by contract, but when the agreement states only that the parties are “joint owners,” these restrictions and obligations may be unexpected.

Clarify the handling of residuals: If your developer requires the ability to reuse knowledge acquired while handling your project, it may insist the agreement include a “residuals” clause permitting the developer to freely use ideas, know-how, and techniques that its personnel retain in their unaided memory. These clauses present two important questions: should the developer's rights in residuals be an exception to the agreement’s confidentiality restrictions, and should these rights constitute a de facto license under your company's IP rights? There are no “right” answers, but to avoid unintended consequences the parties should ensure they address the topic with specificity.

2. Get Specific on Potential Transfers of Rights

Divested entity provisions mean smoother transitions: If your company is obtaining an enterprise-wide license covering its affiliates, including a “divested entity” provision in the agreement will save time and increase value if an affiliate is later sold to an acquirer. These provisions grant licensees the right to use the developed software on behalf of divested affiliates, or allow those former affiliates to continue using the software after the divestiture. Licensees often also insist that the developer be obligated to enter negotiations for a new license agreement with any divested affiliate.

Do not remain silent on sublicensing, transferability, or exclusivity: If a software license agreement is silent on the licensee’s ability to grant sublicenses or transfer its rights, a non-exclusive licensee generally may not engage in either activity without the licensor’s consent. Some courts have extended these rules to exclusive technology licenses. In contrast, a technology licensor typically may transfer its rights without the licensee’s consent. Further, a license agreement that does not state whether it is exclusive or non-exclusive likely will be found to be non-exclusive. Of course, the best practice is to ensure the agreement addresses all these topics in a manner reflecting the parties’ specific intent.
Watch for unexpected restrictions on transfer: A merger transaction in which the target company does not survive the transaction may constitute a breach of non-assignment provisions in the target's agreements for in-licensed IP and software. In part to avoid these restrictions, divestitures often are structured as reverse triangular mergers (RTMs), whereby the target company survives the merger as the buyer's subsidiary. However, a recent court decision has made it less certain that an RTM will not breach non-assignment provisions in the target's commercial contracts. In Meso Scale v. Roche (2011), the Delaware Chancery Court left open the possibility that Roche Holding Ltd.'s RTM acquisition of BioVeris Corp. breached a non-assignment clause in one of BioVeris's agreements. To avoid uncertainty, the prudent approach is to clarify with greater specificity in the development agreement what transactions are and are not permissible.

3. Address Bankruptcy

Acknowledge bankruptcy risk: Section 365(n) of the U.S. Bankruptcy Code protects licensees of patents, copyrights, and trade secrets (but not trademarks) from unilateral termination by bankrupt licensors. Software licensees typically can retain their rights as long as they continue to pay license fees when due and otherwise comply with the agreement. Including an express acknowledgement in a software license that Section 365(n) applies helps avoid disputes if a bankruptcy occurs. If the developer is a significant bankruptcy risk it may be appropriate to insist on a perpetual, irrevocable license to the software.

Get a present license to escrowed source code: Bankruptcy can trigger the release of escrowed source code to a licensee. However, bankruptcy courts may characterize a license contingent on a licensor's bankruptcy as an unenforceable transfer of assets from the bankruptcy estate. Drafting the source code license as a present grant of rights (i.e., “hereby grants”) helps ensure its effectiveness. The developer-licensor may object that it is not allowing your company to presently use the software, but of course the licensee receives no benefit from the license until the escrow agent releases the code.

Bankruptcy complications can be avoided altogether by expanding the scope of escrow release events to include “pre-bankruptcy” warning signs of financial distress, such as licensor's failure to timely pay bills, or concerns expressed by independent financial auditors. Further, a prudent licensee will consider making bankruptcy (and other release events) an exception to any restriction on hiring the licensor's programmers. If a release event occurs, your company may find it difficult to use the source code without assistance from the programmers who know it best.

4. Ensure Ownership is Correctly Transferred

Use the present tense in any assignment of rights: Language intended to assign IP rights to your company should reflect a present transfer (“hereby assigns”), not a future promise to transfer (“will assign” or “agrees to assign”). Under the latter formulation, the developer's failure to deliver the promised assignment may result in a breach of contract claim, but not necessarily ownership of the IP (particularly if the developer already assigned the rights to someone else).

“Work made for hire” is ineffective: Parties often characterize software as a “work made for hire” in development agreements, believing this language will cause copyright ownership to automatically vest in the company, rather than the developer. However, under U.S. copyright law software code generally cannot be a “work made for hire” when created by an independent contractor—even if the parties agree otherwise by contract. This problem can be avoided by including a present assignment of IP rights to your company together with the “work made for hire” language (although under California law including a work-for-hire clause may evidence an employment relationship with the developer and potentially expose your company to employee benefits or workers' compensation claims).

Account for pre-existing rights: If the developer includes any of its pre-existing code or third-party code in developed software intended to be owned by your company, or if a license to other IP rights is necessary to use the software, to avoid IP infringement claims you should ensure the developer identifies those pre-existing rights and grants a perpetual, irrevocable license. In particular, the developer should identify any open source software (OSS) incorporated in the developed software so your company can take precautions to avoid using it in a manner that triggers the "viral" provisions of the applicable OSS license (e.g., prohibitions on collecting royalties for permitting others to use the developed software).

5. Monitor Indemnification Carve-outs

Software developers often indemnify their customers for third-party IP infringement claims, but the exceptions to this obligation warrant careful consideration. In general, a reasonable middle ground is to allocate infringement risk to the party that bears the greatest responsibility for the allegedly infringing conduct. For example, your developer may seek to avoid responsibility for software designed to your company's specifications, but consider narrowing this to instances where non-infringing compliance with your company's requirements is not possible. Similarly, developers may seek to exclude infringement claims relating to combinations of their software with your company's or third-party services or materials, or that relate to post-delivery modifications. However, such carve-outs may be inappropriate when the software would have infringed third-party IP rights notwithstanding the combination or modification.

Daniel C. Glazer is a corporate partner at Fried, Frank, Harris, Shriver & Jacobson and co-head of the firm’s technology transactions group. Mr. Glazer has extensive experience structuring and negotiating complex domestic and cross-border intellectual property and information technology transactions. He regularly advises on IP, technology, and commercial issues for a range of companies—from foundation-stage companies (particularly in New York and the U.K.) to Fortune 50 corporations—in industries such as telecommunications, financial services, consumer products, pharmaceuticals, food and beverages, media and entertainment, and sports. Arianne H. de Govia is a corporate associate at Fried Frank and focuses her practice on the intellectual property aspects of a wide variety of corporate transactions (including mergers and acquisitions, divestitures, financing, and licensing) and also on securing, evaluating, and enforcing intellectual property rights. The authors would like to thank summer associate Daniel Cherif for his assistance in preparing this article.

Reprinted with permission from the February 27, 2013 edition of CORPORATE COUNSEL © 2013 ALM Media Properties, LLC. This article appears online only. All rights reserved. Further duplication without permission is prohibited. For information, contact 877-257-3382 or reprints@alm.com. # 016-02-13-03