Modern software contains a mix of open source, third-party, and proprietary code. They all contribute to the dimensions of risk.

For open source and third-party software, legal risk is real. Over 85% of transactions audited in the Synopsys "Open Source Security and Risk Analysis" (OSSRA) report contained license conflicts. Being out of license compliance can have real-world consequences, including:

- Concerns for buyers when divesting
- Impact on corporate reputation (if sued for licensing, etc.)
- Lawsuits
- Loss of IP (if license compliance requires distribution of source code)

To create a full risk profile of the technology you're acquiring, you must examine its security risk across multiple dimensions. Issues with the architecture, proprietary code, or open source components could lead to points of weakness or entry, making the software vulnerable to attack. The Harvard Business Review notes that in the last several years, concerns about data security and hacking have risen, specifically in M&A.[2] The Black Duck Audit Services team reported that almost 80% of transactions involved code with high-risk vulnerabilities. The consequences associated with such vulnerabilities include:

- Unplanned work to remediate security issues
- Regulatory issues (e.g., GDPR, HIPAA, PCI DSS)
- Attacks by bad actors
- Data breaches (e.g., losses of customer data, intellectual property)

Software quality risk may not have as immediate an impact as a lawsuit or security breach, but it's more insidious. Low-quality code written in a nonmodular way is hard to maintain and significantly reduces productivity in adding features, fixing bugs, and patching vulnerabilities.

Low-quality code represents technical debt—non-value-added activities (e.g., bug fixes, brittle code maintenance, code refactoring)—a burden now that steals resources from the future by reducing developer availability. The impact: Developers working on poorly structured codebases are 60% less productive than those working on well-structured codebases.[3]

There are several best practices for acquirers evaluating software from a legal, security, and software quality perspective.

To surface and mitigate legal problems latent in the codebase under consideration for M&A, you need to identify all components in the codebase, as well as their respective licenses and the obligations and terms of those licenses. This examination should highlight obligations in open source and third-party licenses and any conflicts entailed by the distribution model and licensing for the entire work. It also should determine which components in the codebase require commercial licenses so you can verify that the target has them in place. For completeness, the process must also identify third-party code that may have vague grants of right in code comments, or as is often the case, no applicable license at all. Software licensing can be complex. It's important to involve an open source–savvy lawyer to provide guidance.

A security evaluation should include a look for high-level design flaws, which account for 50% of security vulnerabilities,[4] and ensure security controls are designed into the architecture. Next, it's important to analyze the proprietary code for security bugs—both from the outside-in via penetration testing and from the inside-out with static analysis tools. With open source comprising the majority of many codebases these days, it's also critical to examine open source and third-party components for known

Due diligence can uncover issues that you and the target must address. Some of those issues may warrant technical remediation, with the seller agreeing to fix the issue before or after the close. You can also use language in the definitive agreement to appropriately allocate risk. If you identify a risk that the seller deems a nonissue, the seller may provide a warranty, for example.

Sometimes, you'll want more protection against a risk that surfaces in due diligence—for example, a license issue that could lead to IP litigation, or a major security concern. You and the target can agree to hold back money for you to draw from to cover future costs as needed. Typically, you'll set aside a percentage of the deal value as insurance for a finite time in case issues arise. This percentage can increase if major unexpected issues appear. Extreme surprises that arise could even lead to a lower deal valuation, and though rare, the deal getting derailed altogether.

When you're looking for a vendor to provide an M&A software audit solution, two qualities are table stakes: competent people to work with and good tools. But that's not all. M&A projects fundamentally differ from general IT technical consulting. M&A due diligence imposes special demands that many consultancies cannot meet. So it's critical to find a firm that has M&A experience. And with short due diligence timeframes (e.g., two- to four-week turnarounds), you also need a firm that is:

- **Hyperresponsive.** A deal's pace could be measured in days or hours.
- **Trusted.** Both acquirers, and more critically, sellers should know of and trust the firm.
- **Secure.** The vendor should handle the code for safekeeping.
- **Comprehensive in its services.** With all the loose ends in a deal, it's better to use fewer vendors.

Black Duck® Audit Services comprehensively assesses the software in M&A due diligence. We offer these types of software audits:

- Open source and third-party software audits
- Application security audits
- Software quality audits

To learn more about Black Duck Audits and how we can help you understand software risks, please visit synopsys.com/ open-source-audit.

References

1. 451 Research, Business Impact Brief: Don't Leave Open Source Analysis Out of M&A Due Diligence, 2018.
2. Chirantan Chatterjee and D. Daniel Sokol, Don't Acquire a Company Until You Evaluate Its Data Security, Harvard Business Review, April 16, 2019.
3. Dan Sturtevant, Modular Architectures Make You Agile in the Long Run, IEEE Software 35:1, Jan./Feb. 2018.
4. Gary McGraw, Software Security: Building Security In, Addison-Wesley, 2006.