SYNOPSYS®

WHITE PAPER

Software Due Diligence:

A buyer's checklist and guidelines

Explore the right questions to be sure you understand the risks of the technology you're acquiring

The potential risk and exposure posed by M&A transactions demand thorough and robust due diligence practices. When an acquisition includes important software assets, it's critical to ensure that latent associated risks are identified and fully understood. Knowing what questions to ask when performing due diligence is key to avoiding potential problems and legal complications that could otherwise manifest as post-close surprises. This is especially true for the acquisition of software, which more often than not contains risks that may not be immediately evident.

The Black Duck® Audit Services team has performed thousands of M&A audits to help customers get the full picture of the risk they are taking on when acquiring new technology. Black Duck audits almost always produce findings that, left unidentified or unmitigated, could lead to issues and liability downstream.

What to worry about in M&A transactions

Software due diligence, when performed adequately and effectively, examines the target's code across various dimensions. Some key areas of focus include:

- **Technology**: In technology M&A acquisitions, due diligence activities must incorporate a thorough examination of the technology itself. Since the value of software-centric acquisitions is based on the software and its IP, digging deep for potential quality, legal, and security risks is critical to ensuring the overall worth and viability of the transaction. Assessing the technology in an M&A transaction involves due diligence activities in multiple areas.
    - **Product/strategy:** It's imperative to fully understand the product(s) and strategy of the target. Research on the target market, product roadmap, and competitors provides key insights. Evaluating how the target technology fits into existing portfolios is also important.
    - **People:** Evaluating the people involved in creating and maintaining the software is important to determine areas of strength, potential gaps, and how they will fit into the acquiring company.
    - **Processes/tools:** An evaluation of processes and tools helps determine whether sufficient practices are in place and where improvements or modifications can and should be made. It will also illuminate potential integration issues down the line.
    - **Technology stack:** Good due diligence practices should include an evaluation of the technology itself: its architecture and code provide the foundation for the software. Weaknesses, vulnerabilities, and license obligations can lead to future liabilities and risk if not adequately identified and examined.
- **Legal:** The evaluation of legal risks should include an examination of intellectual property, corporate governance, and antitrust. Failure to perform adequate due diligence can result in lawsuits, a negative impact on business reputation, and loss of IP.
- **Security:** Creating a full risk profile of the target technology is very important. Security risk should be evaluated from the perspective of architecture, proprietary code, and open source code. Failure to do so can result in data breaches, regulatory issues, and unplanned remediation costs and activities.
- **Software quality:** Software quality risks may not be immediately obvious, but they can wreak havoc on future development plans. Low-quality code must be brought up to snuff. Worse, software that isn't organized to be modular and hierarchical will result in future bugs, vulnerabilities, and an overall difficulty in remediating these issues—all a big drag on future development.

The thousands of Black Duck audits performed have uncovered just about every possible issue that might lurk in a codebase or the processes around it. To help guide your software due diligence activities, we have developed a detailed checklist of all the questions to explore when assessing target technology.

Due Diligence Checklist

Product/strategy				
✔	Area of focus	Questions to explore	Why	Who/how
	Strategic fit	Is this product a strategic fit with your current market, portfolio, and growth strategy? What is its current position in the market?	Ensure that the target product addresses a similar buyer and go-to-market motion for ease of integration into the overall company direction. Assess position in the market via SWOT analysis to understand competitive and market pressures and opportunities for growth.	Do it yourself. This is the basis of your investment thesis.
	Product	Does it function as advertised?	It's likely that the product addresses the high-level value proposition, but it may not be mature enough to completely deliver. All products have scars and warts, but you want to ensure that it's at least good enough for the market.	Utilize industry expertise to try it out. It's best to use in-house product/engineering management resources or a consultant with specific industry expertise. Ideally talk to some customers as well.
	Product licensing	How is the product licensed to end users and how are entitlements handled?	This question is key to understanding if the current licensing model will fit with the acquirer's current model and systems. It's important to know if there are potential areas of abuse of the product due to unenforceable entitlements.	An experienced technical leader should be able to answer these questions via interviews with key personnel.
	Roadmap	Does the product roadmap address key customer needs and market trends, and improve market position?	Understanding the product direction will ensure that the target company is positioned to address market needs and capture future revenue growth. It will also help determine if the strategy will need to be refactored post-acquisition.	Utilize product marketing management expertise.

People/organization				
✔	Area of focus	Questions to explore	Why	Who/how
	R&D investment	Is current R&D investment appropriate for the organization?	If R&D is over- or under-invested, especially relative to the market, this might be an area that needs to be addressed once integrated.	Benchmarking against the market is a solid approach.
	Technical leadership	Are the right leaders in place to achieve company goals?	Having the right senior people in place and ensuring they will remain helps provide a smooth transition and aid with employee retention.	You'll want to make your own call, but get input from an experienced technical resource who has a chance to interact with key personnel.
	Crucial employees	Have crucial employees been identified for retention to ensure the success of an integration?	It's important to hold on to crucial employees and understand who may be too crucial. If too much lives only in the technical founder's head, it must be extracted and shared.	Don't just get a list; have an experienced technical resource talk through this with the target's technical leaders.
	Scalability	Is the organization structured and equipped to meet the growth requirements of company goals?	If growth is key, it's important to know that the organization is capable of identifying, bringing on, and integrating new talent.	An experienced technical resource, ideally one who has experience with larger and smaller development organizations, can assess through interviewing technical leadership.
	Team communications	How does the team communicate among itself and with other teams?	Particularly with distributed teams, communications are critical to efficiency. The best organizations prescribe processes and tools to ensure teamwork.	A technical resource experienced with development teams should interview and review documentation.

Process/tools				
✔	Area of focus	Questions to explore	Why	Who/how
	Development process and fit	Is there a well-defined and consistent development process?	Predictable future development is almost always a goal, so a solid development process is critical. If the plan is to integrate with other teams, be sure that the respective processes will mesh.	An experienced technical resource can assess through interviewing technical leadership. Reviewing process documentation is key. If the team is to be integrated, the resource should understand your processes.
	Development process maturity	Are there mature processes in place for planning, development, defect tracking, build, testing, maintenance? Are they documented?	Startup companies may get by with just blood, sweat, and tears, and without having all the boxes checked. In order to scale, investment may be required to fill in deficient processes.	An experienced technical resource can assess through interviewing technical leadership. Reviewing process documentation is key. If the team is to be integrated, the resource should understand your processes.
	Customer requirements management	How does the team learn of, prioritize, and track customer requirements?	A customer-driven roadmap is critical to business growth. Many smaller organizations are dependent on the intuition of a technical founder and might not have formal processes.	An experienced technical resource can assess through interviewing technical leadership. Reviewing process documentation is key.
	Tools	What tools are being used to support development? Will they continue to be used?	Closely tied to process, it's important to see that the team is well-equipped with modern tools that will allow the scaling required.	An experienced technical resource can assess through interviewing technical leadership. Reviewing process documentation is key. If the team is to be integrated, the resource should have knowledge of key tool requirements.
	External tech services	Are there any service agreements in place that will continue post-acquisition? Are there opportunities for contract consolidation?	Agreements for third-party tech services like cloud hosting, payments processing, and the like should be evaluated to determine go-forward costs and opportunities for contract consolidation.	An experienced technical resource can assess through interviewing technical leadership.

Code content, IP, and compliance				
✔	Area of focus	Questions to explore	Why	Who/how
	Open source policy and inventory	Is there a documented policy on open source usage and supporting processes? Can the target produce an accurate open source and third-party code inventory and demonstrate that they are complying with license obligations?	Without a working policy and process, a development organization is unlikely to be able to account for much of their code. This likely indicates latent license issues.	Policy and processes should be reviewed by someone familiar with open source and industry norms. Further, a trusted third party with open source expertise and sophisticated tools is required to verify the contents of the code, and an open-source-specialized attorney should review compliance issues.
	Third-party licenses (commercial and open source)	Are all licenses for third-party software (OSS and commercial) identified and are their obligations met?	If the answer is no, this poses a great risk to intellectual property.	A combination of target disclosures and a code audit will provide a software licensing attorney with what they need to review.
	Encryption	Does any exported code contain strong encryption? Is the proper paperwork in place?	Export compliance requires filling paperwork with regard to encryption.	If the company is well-organized, a lawyer can simply review the paperwork. If it isn't, a trusted third party with analysis tools may be required to identify encryption in the code.
	Web services/ APIs	Does the software rely on any third-party APIs that could pose legal or data privacy concerns?	If the software calls third-party APIs, the company must ensure that terms and services are being met and monitored. It's also important to understand what security or privacy risk they entail. If an API passes sensitive or personally identifiable information without key controls, that data is at risk.	Very few companies track their developers' API use. As this requires source code access, it's best evaluated by a trusted third party armed with analysis tools.
	Existing patents and trademarks	Can the target provide a list of all trademarks, existing patents, patents in progress, and potential patents?	Understanding what parts of the software and brand are protected under trademarks and patents helps determine what has differentiated value in the market vs. what is unenforceable.	An experienced technical leader and legal counsel should be able to provide this information.
	Compliance certifications	Does the target company have compliance certifications for any products (i.e., SOC2, PCI, ISO, and so on)?	Understanding what certifications are being claimed, and validating how they were achieved, as well as having processes in place to retain the certifications, is critical in many markets.	An experienced technical leader should be able to provide certifications and documentation for how they were achieved.

Security				
✔	Area of focus	Questions to explore	Why	Who/how
	Security controls and processes	Are there defined processes for writing secure code? Are there key security controls in place to control for potential weaknesses in password storage, identity/access management, and the like?	Even well-written code must include proper controls in order to be secure. Investment will be required to put these in place.	An application security expert can determine this by examining the architecture from a security perspective.
	Vulnerability assessment	How secure is the software currently? Does it contain security bugs or utilize components with known security vulnerabilities?	Vulnerabilities in the code are subject to breach and must be remediated.	As this requires source code access, it's best evaluated by a trusted third party armed with analysis tools to identify coding bugs and components that may contain known vulnerabilities.
	Penetration testing	Has there been a recent penetration test to ensure the robustness of software security in its running state?	Pen testing is a best practice that enables a company to find security holes before bad actors exploit them.	This should be performed by an expert ethical hacker on a test system. Tests on production systems are suspect because pen testers will "go easy" to avoid breaking the system.

Software quality				
✔	Area of focus	Questions to explore	Why	Who/how
	Code quality	Is the code buggy or poorly written? Are there weaknesses in the code and/or large amounts of technical debt?	Excessive technical debt in the form of bugs means time and effort will be required to ensure code meets industry quality standards.	As this requires source code access, it's best evaluated by a trusted third party armed with analysis tools.
	Duplication	Is the code repetitive either within or between files?	This is an issue with productivity and future development. Best practice is to have minimal duplication so that code changes don't need to be implemented in multiple places, which takes more time and increases the chances of introducing bugs.	As this requires source code access, it's best evaluated by a trusted third party armed with analysis tools.
	Complexity	Is the code over-engineered, difficult to comprehend, or architecturally opaque?	These questions help determine whether key coder "heroes" should be retained to make enhancements, and whether new coders would struggle to contribute.	As this requires source code access, it's best evaluated by a trusted third party armed with analysis tools.
	Commenting/ documentation	Are there sufficient inline comments, both tactical (documenting segments) and strategic (high level), to make the coding clear to new eyes? Does the code owner preserve offline documentation for coding standards and enforce them?	Both the code and how it's written should be documented. When code is clearly documented, it's much easier to bring new developers onto the project, and that means less dependence on past coders (who may no longer be around) and tribal knowledge.	As this requires source code access, it's best evaluated by a trusted third party armed with analysis tools.

Design quality				
✔	Area of focus	Questions to explore	Why	Who/how
	Architectural definition	Is the architecture documented and does the reality conform to the documentation?	Without this discipline, a codebase will easily become unmanageable or may already be.	An experienced architect should evaluate the architecture as documented and dig into how closely the code conforms.
	Organization and structure	Is the architecture well-designed to be modular and hierarchical?	Well-structured code that conforms to a documented architecture is a good indicator that it will be able to rapidly improve functionality. Conversely, poorly structured code means slow future development.	It's possible to determine this through diagrams and interviews, but best practice is to use analytical tools to evaluate. As this requires access to the source code, a trusted third party is typically required.
	Maintainability	Is the architecture a drag on development, and will the software need to be refactored?	It's easy for poorly managed code to become cumbersome to maintain because complex dependencies cause every change to break something else.	Once a software architect has evaluated the architecture using the outlined techniques, they can weigh that assessment against your plans for the product. Refactoring is most justifiable for products that require significant future development.
	Scalability	Is the architecture scalable or will it need to be refactored? Are infrastructure and frameworks fit to purpose and future purpose?	A modular, hierarchical architecture is "table stakes." In addition, if "big" is in the plan for the future, the architecture, infrastructure, and frameworks must be able to support the required dimensions of growth.	An experienced architect familiar with a range of frameworks—and ideally armed with good understanding of the modularity and hierarchy of the architecture—should interview target architects to understand performance and capacity chokepoints.

SYNOPSYS®

The Synopsys difference

Synopsys helps development teams build secure, high-quality software, minimizing risks while maximizing speed and productivity. Synopsys, a recognized leader in application security, provides static analysis, software composition analysis, and dynamic analysis solutions that enable teams to quickly find and fix vulnerabilities and defects in proprietary code, open source components, and application behavior. With a combination of industry-leading tools, services, and expertise, only Synopsys helps organizations optimize security and quality in DevSecOps and throughout the software development life cycle.

For more information, go to www.synopsys.com/software.

Synopsys, Inc.
185 Berry Street, Suite 6500
San Francisco, CA 94107 USA

Contact us:
U.S. Sales: 800.873.8193
International Sales: +1 415.321.5237
Email: sig-info@synopsys.com