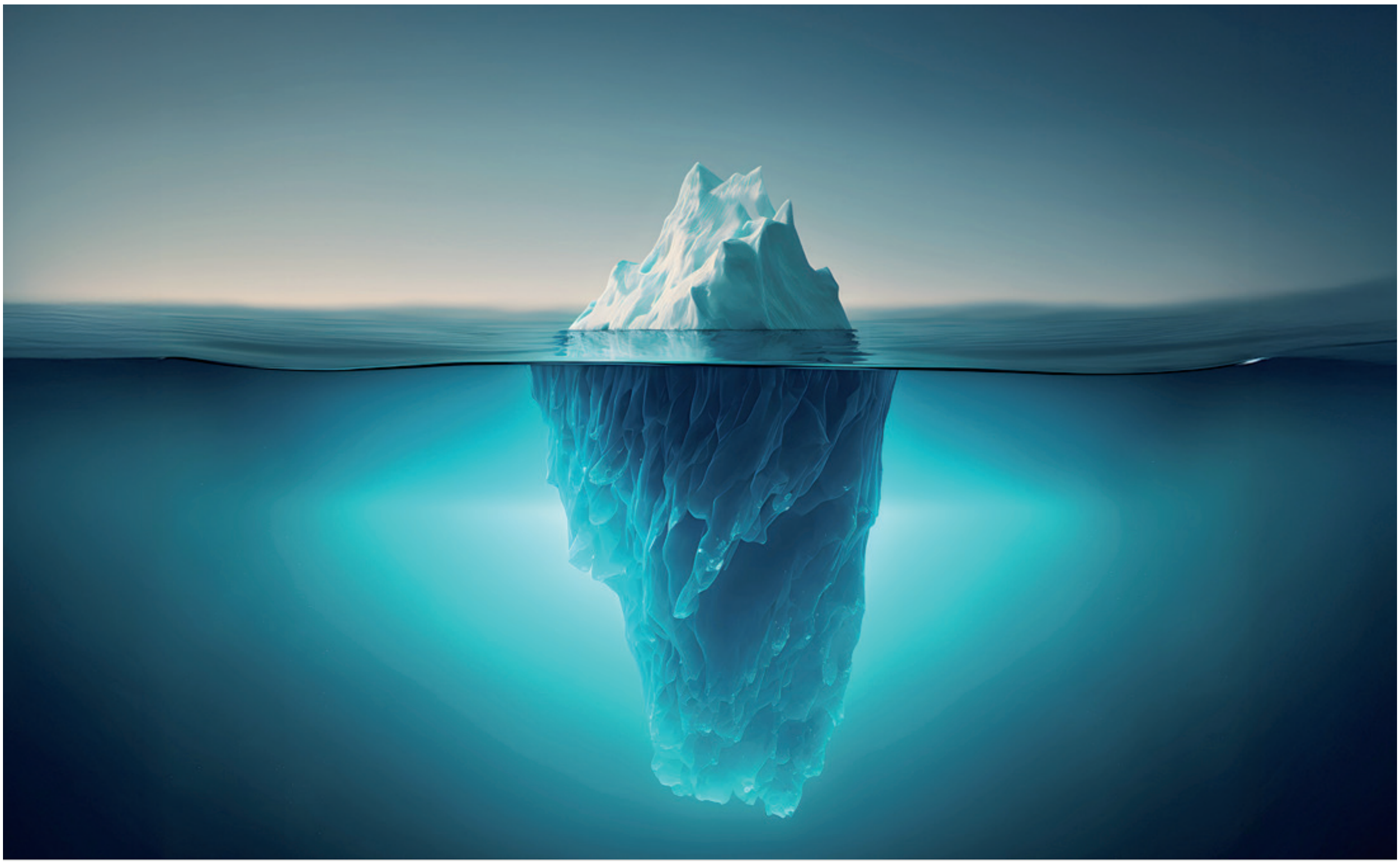
Deloitte.

Powered by CAST



Acquiring a Software Asset : How to Capture Value Without Taking on Debt

Introduction

“A loan is a commitment and must be repaid”: this principle resonates strongly in the context of mergers and acquisitions. Faced with competitive pressure, resource constraints, and the need to accelerate time to market, many companies fall into the invisible trap of tech debt, sometimes without even realizing it. Although tech debt is an inevitable and often unintended consequence of software development, it can reach a critical tipping point and significantly impact the target company’s long-term performance, particularly in terms of innovation and operational flexibility.

The metaphor of tech debt was first introduced in 1992 by American software expert Ward Cunningham. He highlighted a frequently overlooked but essential aspect of software development: technological trade-offs, shortcuts, and other temporary adjustments made to speed up the deployment of a software component must always be considered from a long-term perspective. Like any form of debt, tech debt commits the company that incurs it. Whether related to refactoring, optimization, or bug fixing, the efforts required to address tech debt can have a lasting impact on the health and performance of the target company, sometimes to the point where most of the work becomes remediation rather than innovation.

Understanding and identifying tech debt is therefore essential to ensure the profitability of investments in the tech sector and to maximize the value of M&A transactions. A thorough assessment of tech debt during the due diligence phase, combined with proven technical expertise, allows investors not only to anticipate and mitigate risks, but also to establish a solid foundation for creating long-term value and securing their competitive position in an ever-evolving technology landscape.

Mathieu Correia
Partner - Deloitte Technology M&A

Editorial

This metaphor is therefore nothing new, but it has resurfaced inexorably since the wave of ChatGPT and other language models.

As generative AI has become a driving force in innovative initiatives since 2023, a divide has emerged among software engineering professionals. On one side are the enthusiasts, convinced that these new models will finally help reduce tech debt related to maintenance, documentation, and even automate software refactoring. On the other side are experts who foresee a risk of increased tech debt due to the non-deterministic nature of generative AI and its limitations when dealing with complex software architectures, according to the current state of the art!

What is at stake for companies is not only the construction of brand-new (Greenfield) software, but also the maintainability and safe evolution of existing critical assets (Brownfield). Indeed, generative AI can produce code, but it does not guarantee consistency, compliance, or even the preservation of architectural knowledge, especially if part or all the software is built by this AI!

IT decision-makers and CTOs have thus become aware of both the opportunities and the limitations of AI and are now seeking greater visibility to better manage the tech debt of their software assets. Above all, there is a need for an objective metric to monitor trends and optimize investments.

However, using AI to address software tech debt remains an achievable goal. To this end, we have observed among several of our clients the need to enhance generative AI with a deterministic knowledge base containing an accurate representation of inter-module dependencies. This approach helps to increase the reliability of AI use cases and mitigate some of its limitations.

Greg Rivera
VP Product Management, CAST

Integrating software tech debt into the strategic roadmap

A diverse and complex reality

Insufficient documentation, redundant software components, obsolete dependencies or APIs, tightly coupled components, and ineffective error handling and testing, all these are ways in which tech debt manifests itself, often progressively. Frequent and suboptimal changes to the source code lead to a constant increase in software complexity and disorder. This code entropy, sometimes worsened by high developer turnover, becomes apparent as new features are added or bugs are fixed.

In M&A, this phenomenon becomes even more pronounced: the convergence of heterogeneous systems, the rapid integration of teams, and the need to align different architectures all contribute to the emergence and accumulation of tech debt, often in ways that are difficult to control.

Whether intentional or unintentional, this diversity in form and nature systematically increases the complexity of source code management, making maintenance tasks more arduous and time-consuming. Fixing bugs, developing new features, or simply ensuring application stability thus becomes a delicate exercise.

This complexity is particularly problematic during migrations to new platforms or technology upgrades: the multitude of dependencies, the heterogeneity of architectures, and the obsolescence of certain modules all slow down and complicate these processes. The diversity and accumulation of tech debt ultimately weigh heavily on the teams' ability to evolve the information system efficiently and securely.

Increased maintenance costs and opportunity costs

In 2024, mature organizations dedicate a significant portion of their IT budgets to repaying tech debt in all its forms. The CIGREF (French association of large organizations focused on digital transformation) also highlights that IT debt, in the broad sense, affects all organizations regardless of size and requires proactive governance involving both IT departments and business units. As a result, there is a massive opportunity cost for companies in all sectors due to the time, money, and other resources allocated to managing tech debt instead of creating value.

Beyond this loss of potential gains, tech debt also leads to a series of additional and often avoidable costs. Spending on corrective and adaptive maintenance increases, while refactoring or optimization operations become more frequent and more complex. These necessary interventions tie up teams with low value-added tasks at the expense of strategic projects, contributing to the overall rise in IT operational costs.

Up to 30% of companies' IT budgets are devoted to remediating tech debt in 2024.

Tech debt as a cyber risk vector

Tech debt also introduces recurring operational risks, especially in terms of cybersecurity. Legacy or obsolete systems, which are difficult to update, become vulnerable to exploits due to the lack of regular security patches. Vulnerabilities in source code, the use of APIs or dependencies with publicly known compromises, as well as risky cybersecurity practices, are all aggravating factors. The complexity involved in updating legacy systems delays the implementation of essential updates, prolonging exposure to threats and increasing the risk of incidents such as data breaches, information theft, or ransomware attacks.

Non-compliance with data protection regulations, such as the GDPR, then becomes a tangible threat, exposing the company to legal sanctions. The repetition of such incidents not only weakens the company's reputation, but also its business relationships, thereby substantially impacting its investment thesis.

Tech debt is widely recognized as a major factor contributing to software vulnerabilities. Without proactive management, it can turn critical systems into vulnerable targets, highlighting the urgent need to invest in its identification, prioritization, and remediation to protect the digital infrastructures of tomorrow.

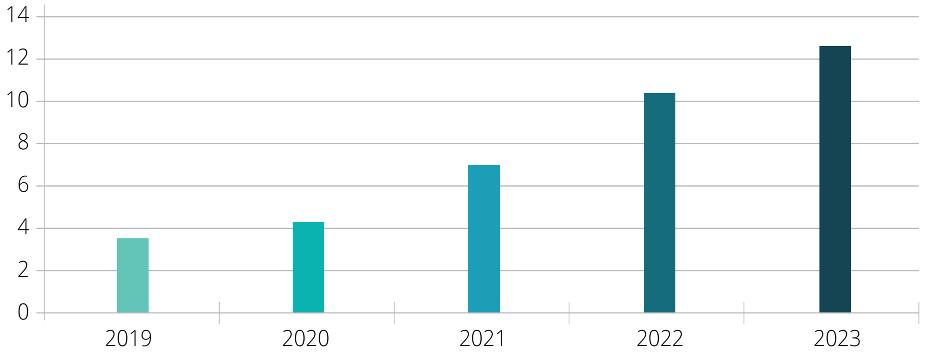
An obstacle to time-to-market and a generator of operational risks

The risks induced by tech debt also translate into strategic risks, such as the loss of innovation capacity and, consequently, a loss of potential for growth and expansion, factors that are critical in the context of an acquisition. Teams, constantly mobilized to manage complexity and address existing issues, have less time and fewer resources to dedicate to developing new features or improving the user experience. This situation hinders organizations' ability to transform and respond effectively to market changes. As a result, it leads to a loss of competitiveness compared to more agile players with less encumbered infrastructures, who can devote most of their efforts to innovation and value creation, thus capturing a significant share of the market in a highly competitive sector. During post-acquisition integration, an obsolete or poorly architected codebase complicates the harmonization of information systems, delays the delivery of new features, and reduces the agility needed to respond to business opportunities.

The presence of obsolete systems and aging infrastructure often involves underestimated investments to restore performance or ensure the compliance of digital assets. This risk directly impacts valuation, as it exposes the acquirer to unforeseen additional costs and a loss of competitiveness if the necessary work is not promptly undertaken. Tech debt can thus undermine the expected profitability of the deal and call into question the ability to realize the synergies identified during the target's assessment.

> "Tech debt is a real debt; sooner or later, it will have to be paid by someone, whether in terms of financial costs, loss of competitiveness, or reputation."

Estimated Financial Losses Caused by Recorded Cyberattacks in Billions of $USD



Source: Internet Crime Complaint Center, FBI – 2023

Identify, prioritize, and manage tech debt for optimal performance

A software asset is composed of two parts: the proprietary code developed and maintained by the company, and third-party components required for its construction and/or use, which are mostly open source.

The proprietary code relies on one or more programming languages, sometimes from different generations, and may interact with other software, whether proprietary or not, such as databases. The main factors contributing to tech debt include: failure to follow best practices specific to these languages, nonexistent or outdated documentation, algorithmic complexity (particularly cyclomatic complexity), excessive coupling between internal components and external dependencies, as well as code size, which is most often measured in thousands of lines of code (kLOC).

As for open-source components, there can be a proliferation of dependencies, both direct and transitive, especially with modern technologies like JavaScript and Python.

The security of these components, their compliance, particularly with respect to intellectual property, and their obsolescence are the main risk factors and, consequently, sources of tech debt.

To identify and manage these risks, CAST offers a dual approach based on the only raw material that matters: the source code. This process begins as early as technical due diligence and continues throughout the software lifecycle, in order to optimize value creation.

Dual approach with CAST software mapping and intelligence

This approach relies on CAST Highlight to see the big picture across an entire software portfolio, and on CAST Imaging to see deep inside the most critical applications.

The approach is particularly well-suited to M&A contexts because it does not require the disclosure of source code outside the company's security perimeter. Furthermore, it is non-intrusive: no agent needs to be inserted in runtimes, which avoids any disruption to the production environment of the software. That is especially important for SaaS solutions serving a large number of clients.

Prioritization within the software and application portfolio

CAST Highlight assesses tech debt using several key indicators:

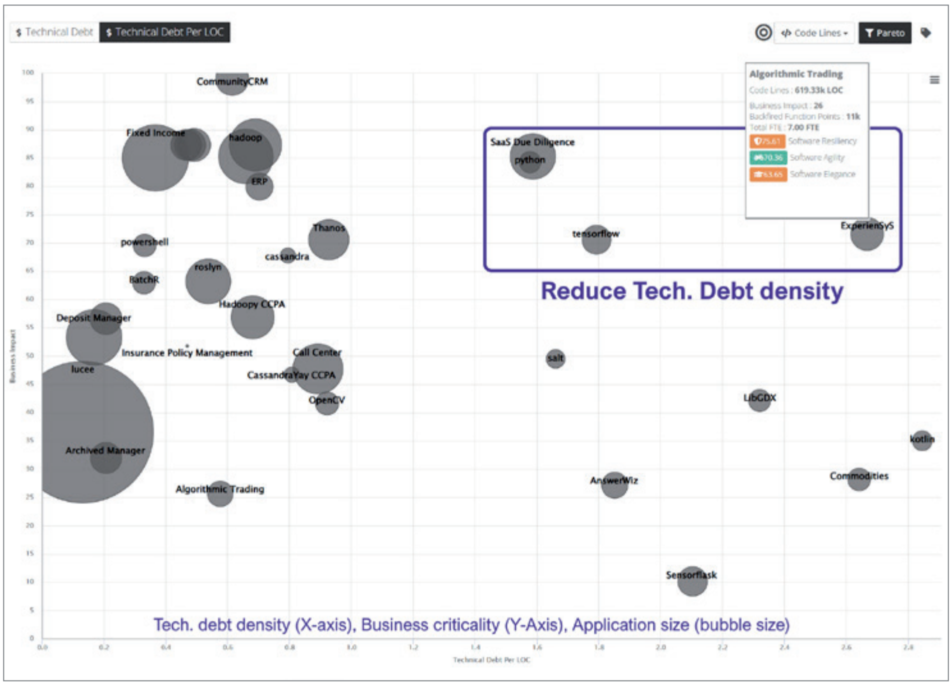
- Tech debt: An estimate of the effort required to address deviations from programming language best practices (C/C++, .NET/C#, Java, JavaScript, Python, SQL, etc.).
- Open Source Safety: Measures risks related to open-source libraries in terms of security, compliance, and obsolescence.
- Cloud maturity: Indicates how ready and optimized each application is for the cloud, pinpointing blockers, required effort, and best-fit cloud-native services.

These indicators provide a relative comparison of applications based on several criteria, including their business impact.

By combining this indicator with the average tech debt density per application, CAST Highlight makes it possible to prioritize applications that are both the most critical and the most at risk (the first quadrant of the graph below).

The volume of code changes over time: it increases with the addition of new features and can decrease through refactoring. However, every line of code that is modified as well as every third-party component can contribute to tech debt. This is why tracking debt density is more useful as a performance indicator than simply tracking the total tech debt.

In this way, during the technical due diligence phase, risks and remediation costs are identified, while in the post-acquisition phase, their evolution is monitored to optimize the action plan.



Source : CAST Highlight

Targeting the Most Critical Software

For major software assets, the approach relies not only on CAST Highlight but also on CAST Imaging for an in-depth analysis based on the ISO 5055 standard.

CAST Highlight pinpoints cloud blockers and risks related to open-source libraries, such as potential intellectual property violations and known vulnerabilities that could be exploited by malicious actors. It also generates an accurate, infinite-depth SBOM (Software Bill of Materials), key deliverable during technical due diligence.

CAST Imaging, through its Gatekeeper extension, surfaces structural flaws in the system construction across all layers and technologies comprising the application by evaluating it against ISO 5055, the industry's most stringent standard.

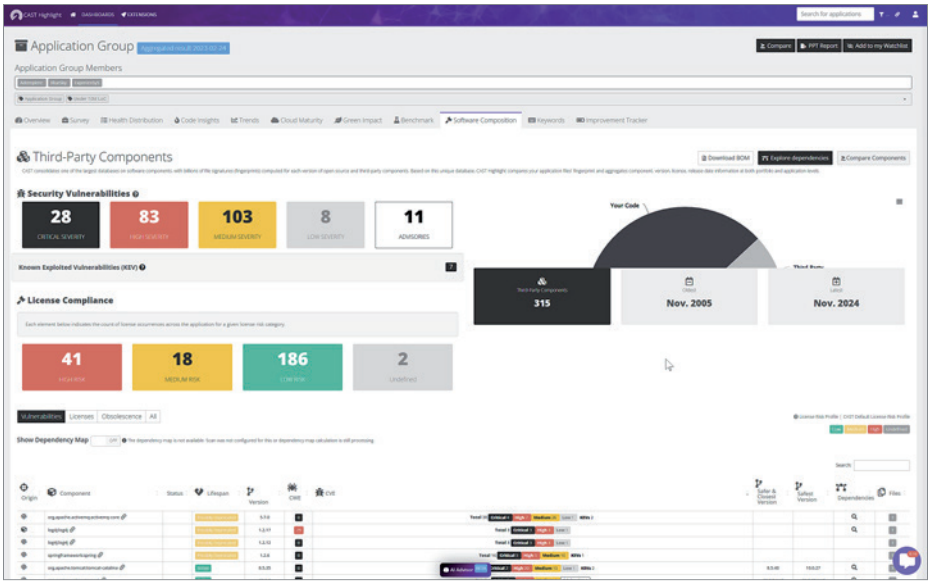
Structural defects typically represent about 8% of all defects in the code, yet result in 90% of the significant security, reliability, maintainability, and performance efficiency issues, i.e. have the highest negative impact on the business. The analysis allows you to prioritize this critical 8% in order to align the action plan with investor priorities and optimize the business case—by focusing efforts, for example, on defects that could lead to customer data leaks or SQL injection vulnerabilities.

In addition, CAST provides benchmarks for applications in the same industry or with the same technological architecture.

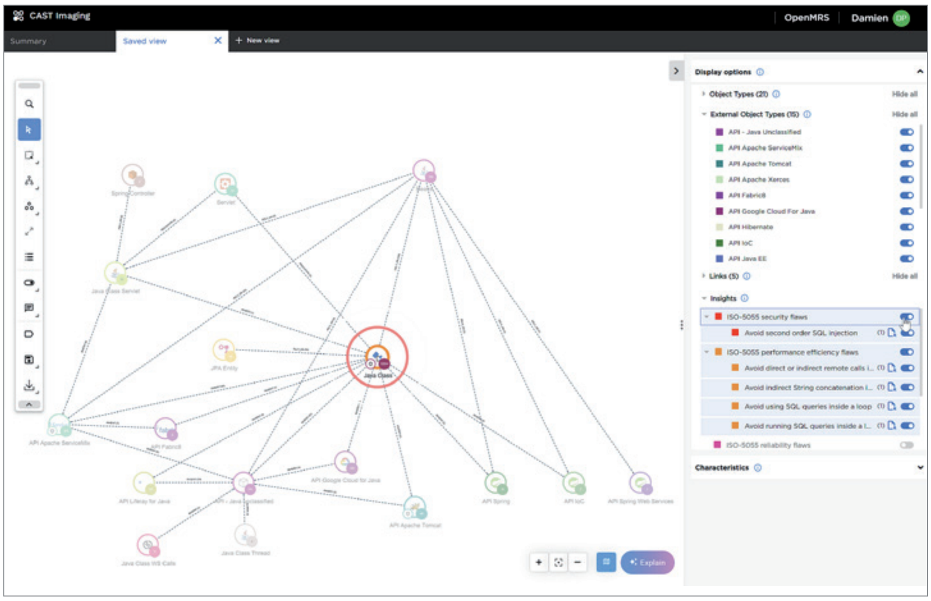
CAST Imaging also enables reverse documentation of the software's architecture from its source code and database schema. This approach represents the software as a knowledge graph, where defects are prioritized based on their impact within the context of the software's architecture.

This representation makes it possible to reliably estimate remediation or transformation costs. For example, it can support decisions such as updating or replacing (carve-out) a risky library, or carrying out the technical modernization needed to shift the business model toward SaaS.

For major software assets, the approach relies not only on CAST Highlight but also on CAST Imaging for an in-depth analysis based on the ISO 5055 standard.



Source: CAST Highlight



Source: CAST Imaging

Additionally, assessing the quality of code documentation is essential, whether for future transformation or simply for daily maintenance. The Maintainability Index considers not just the volume of comments, adherence to naming conventions, and object size, but also the number of parameters, the degree of coupling between objects, and algorithmic complexity.

This index reflects how difficult it would be for a newcomer to understand the code. If this score is significantly below the market average, and there is also a risk of a low bus factor (i.e., too few people understanding critical parts of the system), CAST Imaging can serve as a reliable knowledge base for the technical architecture, both post-acquisition and throughout the transformation journey.

Additionally, assessing the quality of code documentation is essential, whether for future transformation or simply for daily maintenance.

ISO 5055 Standard

Introduced in 2021, the ISO 5055 standard defines engineering rules for evaluating software based on the structural characteristics of its source code. It serves as a framework for estimating architectural tech debt. Developed by the Object Management Group (OMG) and Carnegie Mellon University, these rules focus on the most critical factors for security, reliability, performance efficiency, and maintainability. While only about 8% of defects are structural, they account for 90% of production issues most impacting the business (source: OMG).

Measuring adherence with ISO 5055 therefore helps assess risks related to code architecture.

With the exponential increase of AI-generated code, assessing the structural integrity of the entire application against ISO 5055 is becoming essential.

Risk management for signing and closing

From technical findings to operational impacts

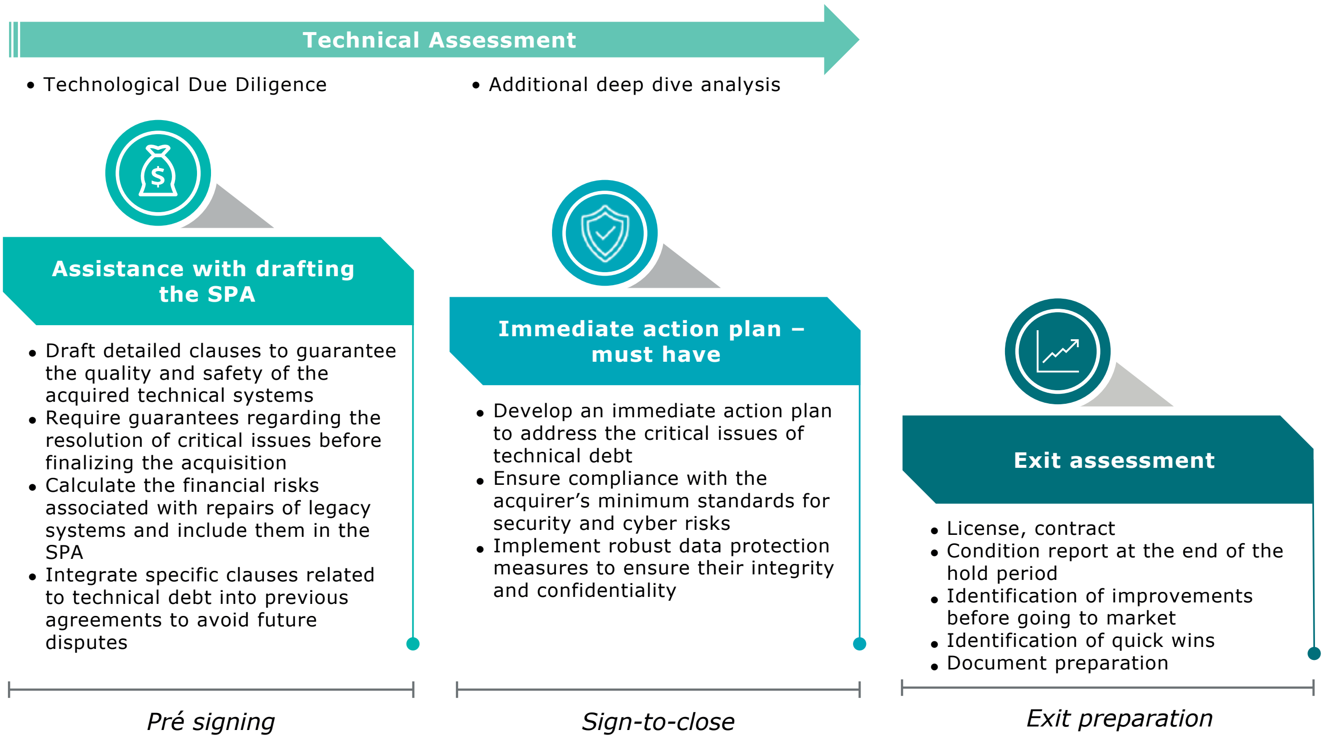
The consideration of tech debt must be integrated into a broader transaction process, from the technological due diligence conducted before signing to the implementation of corrective actions before closing. The use of specialized tools tailored to M&A contexts, combined with the technical expertise of specialized teams such as Deloitte's, enables not only the identification of tech debt, but, more importantly, its prioritization and management within a demanding acquisition environment, often under restrictive intervention constraints. Indeed, the issue of tech debt extends beyond purely technical or engineering aspects. Above all, it is a valuation challenge that can affect the value of a target company, especially when the debt is linked to clearly identified operational risks.

The strength of the investment thesis depends directly on the company's ability to identify and effectively manage this issue: evaluating acquired digital assets, particularly in terms of scalability and flexibility, becomes crucial. Conversely, a proactive approach to managing tech debt, through targeted improvement and optimization strategies, can turn this challenge into a real opportunity, strengthening the robustness and innovation of information systems. This proactive stance supports value creation, especially during the retention phase, by maximizing operational efficiency and positioning the company for sustained and durable growth.

For investors, it is therefore essential to analyze how a company addresses the issue of tech debt. Investing in an organization with a clear strategy for managing this debt, supported by a detailed plan for its reduction or restructuring, can be a positive indicator of the company's ability to achieve sustainable growth and maximize long-term profitability.

The development of a detailed remediation plan makes it possible to precisely define the actions required to address the identified technical issues. Implementing tracking metrics is then essential to assess progress and ensure that the highlighted issues are effectively resolved. Involving the target company's technical teams in this process is crucial, as they possess in-depth knowledge of the existing systems.

The identified elements of tech debt should be incorporated into the Share Purchase Agreement (SPA) to protect the buyer and ensure a fair negotiation. Including specific warranty clauses regarding the quality and maintenance of IT systems helps to mitigate the identified risks. It is also wise to provide for escrow funds to cover the costs of necessary remediations. Additionally, stipulating conditions precedent in the SPA, requiring the resolution of certain technical issues before the transaction is finalized, ensures that these issues are addressed before closing, thereby minimizing post-acquisition risks.



Technical Assessment

- Technological Due Diligence
- Additional deep dive analysis

Assistance with drafting the SPA

- Draft detailed clauses to guarantee the quality and safety of the acquired technical systems
- Require guarantees regarding the resolution of critical issues before finalizing the acquisition
- Calculate the financial risks associated with repairs of legacy systems and include them in the SPA
- Integrate specific clauses related to technical debt into previous agreements to avoid future disputes

Pré signing

Immediate action plan – must have

- Develop an immediate action plan to address the critical issues of technical debt
- Ensure compliance with the acquirer's minimum standards for security and cyber risks
- Implement robust data protection measures to ensure their integrity and confidentiality

Sign-to-close

Exit assessment

- License, contract
- Condition report at the end of the hold period
- Identification of improvements before going to market
- Identification of quick wins
- Document preparation

Exit preparation

Technical experts involved as early as possible in the transaction cycle

In a context where technology assets are becoming strategic value drivers, tech debt can undermine the expected synergies of a deal or jeopardize a carve-out.

The Shift-Left paradigm is increasingly recognized as a best practice. Too often, IT and software analysis is postponed to the later technical stages of a transaction, when in fact it should occur as early as the strategic screening phase, or even in support of the investment thesis. Involving a technology expert in the initial phases of a deal enables early identification of weak signals and critical risk areas. This early positioning allows for a cross-functional understanding of both business and IT challenges, improves alignment with value creation strategies, and helps anticipate remediation costs.

By integrating the technological dimension into the transaction process from the outset, investors gain a decisive competitive advantage and significantly reduce post-deal execution risks.

This approach makes it possible to identify issues that are not visible in traditional financial indicators and to turn tech due diligence into a strategic tool. It supports better risk assessment, more accurate pricing, and smoother post-deal execution.

Shift-left

Shift-Left is a strategy adopted in software development that involves moving testing and quality controls to the very beginning of the development process, rather than performing them at the end. This approach makes it possible to detect and fix errors earlier, which significantly reduces costs. By identifying problems at the earliest stages, teams can avoid expensive late-stage revisions or corrections. This not only reduces the expenses associated with fixing defects, but also cuts down on production delays, ensuring faster and more efficient project delivery. By encouraging closer collaboration between developers and testers from the outset, Shift-Left improves the overall quality of the final product while optimizing the investment of time and resources.

By integrating the technological dimension into the transaction process from the outset, investors gain a decisive competitive advantage and significantly reduce post-deal execution risks.

Key takeaways

- Tech debt, in all its forms, represents a major opportunity cost for companies, absorbing a significant portion of resources that could otherwise be devoted to innovation and growth. If left unmanaged, it increases complexity, maintenance costs, and cyber and operational risks, while hindering organizations' ability to evolve and remain competitive in their markets.

- Effective management of tech debt involves identifying, prioritizing, and quantifying risks during the due diligence process, using solutions such as CAST Highlight and CAST Imaging. This approach allows companies to focus efforts on the most critical applications and vulnerabilities, optimize action plans, and ensure reliable remediation cost estimates, while facilitating transformation and value creation post-acquisition.

- The early integration of tech debt analysis and management into the due diligence process is essential to secure the value of a merger or acquisition. A proactive and structured approach, involving technical experts from the earliest stages of the deal, not only helps better identify, prioritize, and address technical and operational risks, but also maximizes value creation, optimizes negotiation, and limits post-acquisition risks.

Glossary

Bus Factor: Risk related to the concentration of knowledge. It refers to the minimum number of key people whose loss would jeopardize the maintenance of the software.

Cyclomatic Complexity: The number of independent paths during the execution of a program or object. For example, nested tests increase this and make the code harder to understand and change in the future.

Tech debt Density: Tech debt relative to the size of the source code, expressed in lines of code. As an indication, the industry average is between 0.63 and 1.64 minutes per line, according to a study conducted by CAST in 2023 on a sample of more than 3,000 software applications representative of different market segments and programming languages.

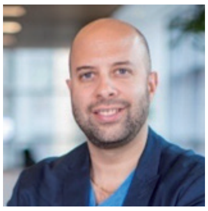
Knowledge Graph: A graph representing the interactions and dependencies within the software and with other software. It is a network of nodes (functions, methods, tables, etc.) and their relationships. Each node and relationship has intrinsic properties such as size, cyclomatic complexity, etc.

SBOM (Software Bill of Materials):
A structured inventory of all software components (libraries, dependencies, modules, etc.) that make up a software asset.

Contacts

Mathieu Correia
Deloitte
Partner | Technology M&A
mcorreia@deloitte.fr

Matthieu Quercia
Deloitte
Consultant | Technology M&A
mquercia@deloitte.fr



Greg Rivera
CAST
VP Product Management
g.rivera@castsoftware.com

About **Deloitte**

Deloitte refers to one or more of Deloitte Touche Tohmatsu Limited ("DTTL"), its global network of member firms, and their related entities (collectively, the "Deloitte organization"). DTTL (also referred to as "Deloitte Global") and each of its member firms and related entities are legally separate and independent entities, which cannot obligate or bind each other in respect of third parties. DTTL and each DTTL member firm and related entity is liable only for its own acts and omissions, and not those of each other. DTTL does not provide services to clients. Please see www.deloitte.com/about to learn more. In France, Deloitte SAS is the member firm of Deloitte Touche Tohmatsu Limited, and professional services are rendered by its subsidiaries and affiliates.

6, place de la Pyramide – 92908 Paris-La Défense Cedex

Designed by dot.

About CAST

Businesses move faster using CAST technology to understand, improve, and transform their software. Through semantic analysis of source code, CAST produces 3D maps and dashboards to navigate inside individual applications and across entire portfolios. This intelligence empowers executives and technology leaders to steer, speed, and report on initiatives such as technical debt, GenAI, modernization, and cloud. As the pioneer of the software intelligence field, CAST is trusted by the world's leading companies and governments, their consultancies and cloud providers.

For more information, visit www.castsoftware.com/