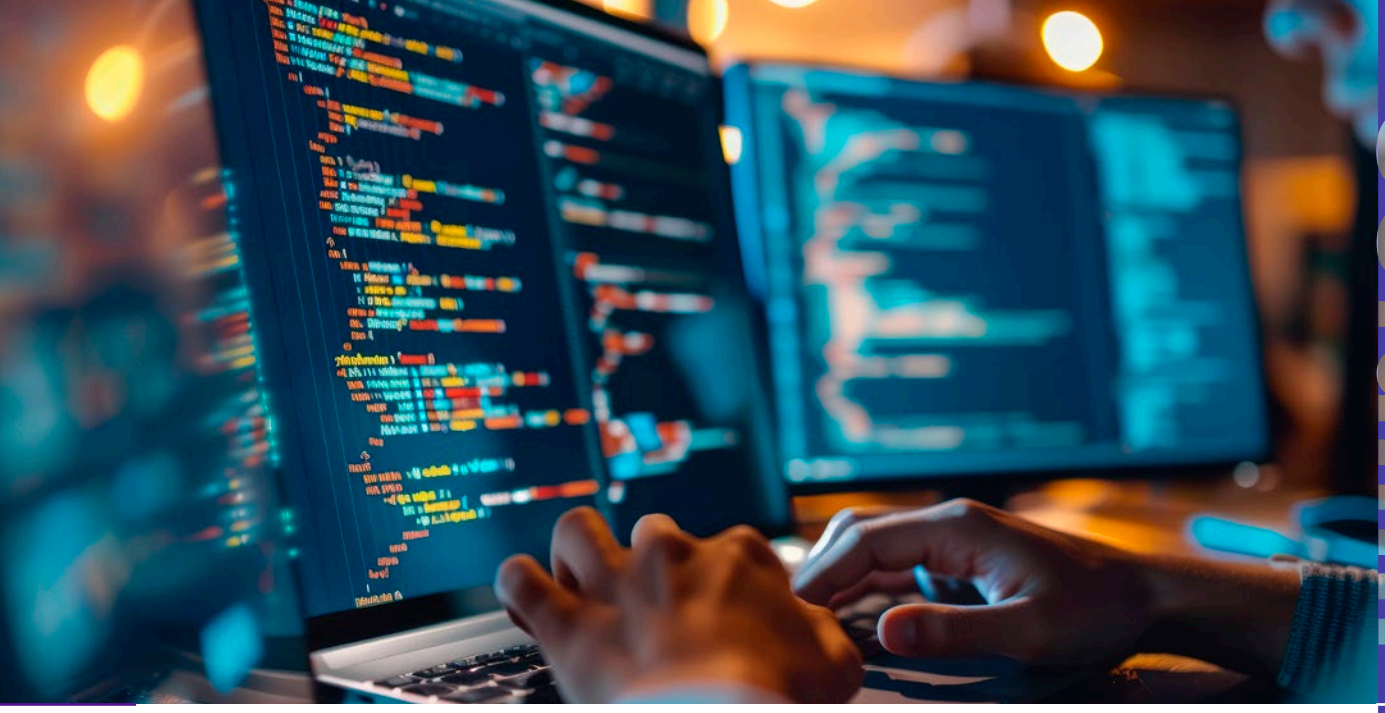

Column

The importance of escrow and source code analysis for software continuity

In the fast-paced world of IT, ensuring the continuity and maintainability of software is paramount. With the rise of new technologies and the reliance on software vendors such as SaaS (Software as a Service), having robust agreements in place with these suppliers has never been more critical. New regulations, such as DORA, align with this trend and demand organizations to be more in control to ensure the continuity of IT assets.

Escrow agreements are a widely used strategy to mitigate the specific risk of service disruption. These agreements provide clients with access to a copy of an application's source code if the software supplier is no longer able to maintain the software. While this "release" of the source code enables clients to compile the solution and maintain continuity in their operations, it also presents a challenging choice: either initiate the procurement process for a replacement product or take on the responsibility of maintaining the software internally or with a new third party. Taking on the responsibility of maintaining the code requires careful upfront planning to ensure a smooth transition while maintaining stable operations. One effective strategy to achieve this is through escrow agreements combined with thorough source code analysis of the vendor's software. This approach not only safeguards the interests of all parties involved but also ensures that the software can be effectively maintained and continued in the event of unforeseen circumstances.

ESCROW AGREEMENTS: A SAFETY NET

An escrow agreement is a legal arrangement where the source code of a vendor's software application is deposited with a third-party escrow agent. This agent holds the code in trust and releases it to the licensee under specific conditions, such as the vendor going out of business, failing to meet contractual obligations, or discontinuing support for the software. This mechanism provides a safety net for the licensee, ensuring that they have access to the source code if the vendor can no longer support the software.

Kevin Bankersen
(Compact editor)

THE ROLE OF SOURCE CODE ANALYSIS

While escrow agreements provide access to the vendor's source code, the real challenge lies in ensuring that the code is usable and maintainable. This is where source code analysis becomes essential. By examining the vendor's code, source code analysis helps identify potential issues, evaluate its structure, and assess its overall quality. This process is essential for several reasons:

1. *Code quality and maintainability.* Source code that adheres to better practices and coding standards is easier to understand, modify, and extend, thus supporting transfer and future maintenance.
2. *Documentation and knowledge transfer.* Documentation explaining the intent, build intricacies, and design principles is invaluable for new developers who may need to work on the software in the future. This is the moment to determine if the documentation can aid in a potential future release.
3. *Compliance and legal assurance.* Ensuring that the vendor's source code complies with relevant regulations and standards is critical, especially in industries with stringent compliance requirements. Source code analysis provides the necessary assurance that the software meets these standards.
4. *Open-source licenses.* Software rarely consists of bespoke source code – many solutions make use of a multitude of open-source packages to support basic functionality. To maintain the solution, having a "software bill of materials" (SBOM) and understanding the current support on these components is mandatory.
5. *CI/CD pipelines.* Additionally, the transfer of Continuous Integration/Continuous Deployment (CI/CD) pipelines is crucial for maintaining the automated processes that support software development and deployment.

Escrow agreements are a good first step to mitigate continuity risks but require a more thorough approach to ensure a robust strategy that protects the interests of all stakeholders. A combined approach with source code analysis and documentation reviews not only provides a legal framework for accessing the source code but also ensures that the code is of high quality, secure, and well-documented. Navigate the complexities of modern technology landscapes with confidence and resilience!