IT implementation and development projects: key trends and legal considerations

PLESNER

IT implementation and development projects: key trends and legal considerations

As organisations increasingly invest in digital transformation, IT implementation and development projects remain a critical area of strategic focus. Whether deploying an enterprise resource planning (ERP) system, developing bespoke software, or migrating to cloud-based infrastructure, these projects carry significant legal and commercial risk. This newsletter highlights current market trends and outlines some of the main legal considerations that parties should address before initiating such projects.

Plesner is hosting a **seminar on 4 June 2026** where we will take a deeper dive into these and many other critical topics shaping IT implementation and development projects today. Spaces are limited — secure your seat and sign up now to ensure you don't miss out.

Sign up here

Current Trends and key takeaways

- **Cloud first-strategies and SaaS-based delivery models.** Cloud-first strategies and SaaS-based delivery models have become an increasingly dominant posture for many organisations. While migration to the cloud remains ongoing in several sectors, the focus is progressively shifting towards optimisation, governance, and the management of complex, hybrid cloud environments. This evolution alters the risk profile of IT projects. Organisations must address issues such as data residency and sovereignty, dependency on critical service providers, vendor lock-in and exit complexity, shared responsibility for security and compliance, and the long-term financial implications of consumption-based pricing models.
- **Increased regulatory frameworks and "compliance by design".** Regulatory frameworks governing data protection, cybersecurity and AI — including the EU's DORA, NIS2 Directive, Cyber Resilience Act, Data Act and AI Act — are imposing enhanced obligations on organisations undertaking IT projects, particularly in regulated sectors such as financial services and healthcare. IT projects should be conducted according to a compliance-by-design principle, meaning that regulatory and legal requirements are embedded at the design stage rather than validated retrospectively. This requires early involvement of legal and compliance stakeholders to review and challenge the system's target design (or "blueprint"), reducing the risk of costly redesign, implementation delays, and regulatory non-compliance.
- **Agentic AI and licence compliance.** As organisations increasingly deploy Agentic AI, AI agents, bots, and automated processes that interact with licensed enterprise software environments, they must carefully assess whether such use is compatible with existing licence

agreements. Many enterprise software products are licensed on a named-user or concurrent-user basis, and the introduction of non-human actors that autonomously access, query, or trigger transactions within such systems may fall outside the scope of existing entitlements — potentially exposing the organisation to material licence breaches and substantial financial liability. While the underlying issue is not new, it has become increasingly acute given the rapid growth of the market for agentic AI tools and services.

- **AI code-generating tools and intellectual property.** The increasing use of AI-powered code-generating tools (such as GitHub Copilot and similar LLM-based coding assistants) in software development projects raises novel intellectual property considerations. Key issues include: (i) the risk of third-party IP infringement arising from tools trained on open-source code with restrictive licence terms; (ii) uncertainty as to whether copyright protection subsists in AI-generated code, given the human authorship requirements in most jurisdictions; and (iii) the need for robust contractual safeguards — including clear IP allocation, broad assignment language, confidentiality obligations, and supplier warranties regarding human review and creative input — to ensure adequate legal protection over code forming part of the delivered solution.

Some key legal considerations

Overall responsibility and delivery models

Where an IT project involves multiple external suppliers, system integrators, or service providers, the question of overall responsibility becomes materially more complex. In a multi-vendor environment, there is an inherent risk that accountability gaps arise at the interfaces between each supplier's respective scope of delivery, with each party disclaiming responsibility for failures attributable to interdependencies or integration points. We see various delivery models addressing these risks. The choice of delivery model, as discussed below, must be assessed not only in the bilateral context of each supplier engagement, but also in light of the overall multi-vendor architecture and the customer's capacity to manage residual integration risk.

A fundamental threshold question in any IT project is the nature and extent of the supplier's responsibility. The contractual spectrum ranges from pure staff augmentation - where the supplier provides personnel to work under the customer's direction and management, with the customer retaining full responsibility for outcomes - to a full delivery model incorporating a specific performance guarantee, where the supplier assumes end-to-end accountability for delivering a defined result that meets agreed specifications.

Between these poles lie various hybrid arrangements, however, the choice of model has profound implications for risk allocation, governance structures, and recourse in the event of project failure. Parties should ensure that the contract reflects the intended allocation of responsibility.

Commercial considerations and payment structure

As with the delivery models discussed in clause 1.2.1 above, the commercial structure of IT implementation and development projects typically falls along a spectrum. At one end lies a pure time-and-materials (T&M) model, under which the customer bears the entirety of the cost risk and pays for actual time expended at agreed rates. At the other end lies a fixed-price model, under which the supplier assumes full responsibility for delivering the agreed scope within a defined budget, absorbing the risk of any cost overruns.

In practice, as with the allocation of delivery responsibility, various hybrid arrangements are common. A single project may combine fixed-price elements (for example, in respect of

well-defined deliverables or work packages where the scope is sufficiently mature) with T&M-based elements (for example, in respect of phases where requirements remain uncertain or are expected to evolve). Furthermore, T&M pricing models themselves exist in a range of variants. These span from a pure T&M arrangement with no cost ceiling, to models based on a fee estimate provided by the supplier, under which the supplier assumes a degree of cost risk in the event that actual expenditure exceeds the estimate - for example, by applying increased reductions to the applicable hourly rates for time incurred beyond the estimated threshold. Parties should carefully consider which pricing structure, or combination of structures, is appropriate for each element of the project, having regard to the maturity of the relevant requirements and the desired balance between cost certainty and flexibility.

When structuring the payment mechanism, customers should normally focus on the timing at which they will derive value from the system. This is sometimes referred to as the "hockey stick". This refers to the characteristic curve describing when the customer will realise the benefit of the system: during the earlier phases of the project, the customer derives little or no operational value from the work performed, with the substantial benefit materialising only upon final acceptance and go-live, when the system becomes available for productive use. This creates a value curve that, when depicted graphically, resembles a hockey stick laid on its side. Customers should consider whether the payment structure should be aligned with this benefit curve, such that the main portion of the total fees is released only upon final approval of the system - i.e., at the point when the customer can in fact utilise and derive value from the delivered solution. A payment structure that is front-loaded or distributed evenly across project milestones may expose the customer to the risk of having paid a disproportionate share of the contract value before any tangible benefit has been received, which may in turn affect the commercial dynamics between the parties in the event of project delays, deficiencies, or delivery failures in the later stages.

Clarification of customer needs, gap analysis and the definition of "standard"

One of the most frequent sources of IT project disputes arises from inadequate specification of the customer's requirements and a failure to conduct rigorous gap analysis at the outset. Where a supplier proposes to deliver a "standard" or "out-of-the-box" solution, the parties must clearly define what "standard" means in the context of the specific engagement - including which functionalities are included, which require configuration, and which necessitate bespoke development.

This phase is further complicated by a fundamental asymmetry of knowledge between the parties. The supplier, while possessing deep expertise in the capabilities and architecture of its own system, will typically have limited understanding of the customer's existing business processes, operational workflows, and specific organisational needs. Conversely, the customer, while intimately familiar with its own processes and requirements, will generally have little insight into the full capabilities, constraints, and configuration options of the new system. This information gap creates a significant risk of misalignment between the solution delivered and the customer's actual needs.

Importantly, the customer should seek to negotiate a contractual right to exit the engagement at the conclusion of the clarification phase, if the gap analysis reveals fundamental misalignments between the customer's requirements and the system's capabilities.

Critically, customers must recognise that a successful gap analysis requires substantial engagement from the customer's own organisation. Sufficient time and internal resources should be allocated to support this phase, and customers should assess - having regard to their own organisational maturity and technical capability - whether external advisory support is

required to assist with the articulation of requirements and the validation of the supplier's findings. Without such clarity, customers risk discovering late in the project that critical functionality requires costly customisation that was not contemplated in the original scope or pricing.

Intellectual property and code-generating tools

Customers should give careful consideration to the allocation of intellectual property rights at the outset of any IT implementation or development project. The appropriate approach will depend on the strategic significance of the deliverables to the customer's business. Where the system or its bespoke components are expected to confer a competitive advantage on the customer, the customer should seek to secure proprietary ownership of the relevant intellectual property, thereby preventing the supplier from making the same or similar solution available to competitors.

Conversely, where the deliverables are not competitively sensitive - for example, where they comprise standard integrations, configurations, or modules of general utility - customers may benefit from permitting the supplier to retain ownership and to resell or reuse the deliverables (or elements thereof) to other clients. Such an arrangement can provide the customer with meaningful commercial leverage to negotiate reduced development costs, on the basis that the supplier will be able to amortise its investment across multiple engagements. In either case, the contract should clearly delineate background IP, foreground IP, and third-party IP, and should specify the rights granted to each party in respect of each category.

As AI-powered code-generating tools (such as GitHub Copilot, Amazon CodeWhisperer, or similar large language model-based coding assistants) become increasingly prevalent in software development, customers should proceed on the assumption that suppliers may seek to utilise such tools in the performance of the development work. Rather than treating this as a binary question of permission or prohibition, customers should address the use of AI-powered code-generating tools expressly in the contract in order to ensure that AI-powered code-generating tools are used in an appropriate manner.

From a commercial perspective, the use of AI-powered code-generating tools can materially accelerate development timelines, reduce the volume of manual coding effort required, and thereby contribute to lower overall project costs. Such tools may also assist in producing more standardised and consistent code, reducing the incidence of certain categories of human error.

At the same time, customers should be aware that AI-generated code may introduce quality and security risks, as the output of such tools is not always predictable and may contain vulnerabilities, inefficiencies, or errors that are not immediately apparent upon review. These risks underscore the importance of establishing robust contractual safeguards rather than leaving the supplier's use of such tools unregulated.

The use of AI-powered code-generating tools also gives rise to an elevated risk of third-party intellectual property claims. These tools are typically trained on large volumes of publicly available source code, including code published under open-source licences with varying degrees of restriction (such as copyleft obligations under the GNU General Public License, or attribution requirements under the MIT or Apache licences). There is a risk that AI-generated output may reproduce or closely resemble code from the training data, potentially triggering licence obligations or infringement claims of which neither the supplier nor the customer may be immediately aware. The difficulty is compounded by the fact that, unlike a human developer who may consciously reference or copy existing code, AI-generated output provides no attribution or provenance information, making it inherently difficult to identify

whether the output is derived from, or substantially similar to, protected third-party works.

A related and increasingly significant legal issue concerns the ownership of intellectual property rights in AI-generated code. In many jurisdictions, copyright protection requires a threshold of human authorship or original intellectual creation, and it remains unsettled whether code generated autonomously or semi-autonomously by an AI tool satisfies this requirement. The position varies across jurisdictions and has not yet been definitively resolved by the courts in most major legal systems.

Where there is a risk that IP rights may not subsist in AI-generated code, parties should consider supplementary contractual mechanisms to protect their position. These may include broad assignment language and firm confidentiality obligations including also the AI-generated code - so that, even in the absence of IP protection, the agreed owner retains practical control over the use and dissemination of the code. Customers may also wish to require the supplier to warrant that AI-generated code has been subject to sufficient human review, modification, and creative input so as to maximise the prospect of copyright protection subsisting in the final deliverables. By addressing these matters proactively, parties can mitigate the risk of being left without adequate legal protection over code that forms a material part of the delivered solution.

Agentic AI and compliance by design

A further area of increasing legal and commercial significance concerns the interaction between AI technologies in particular Agentic AI - and the licence models applied by major enterprise software vendors such as Microsoft, Oracle, and SAP. As organisations seek to deploy Agentic AI that interact with licensed software environments, they must carefully assess whether such use is compatible with the terms of their existing licence agreements. Failure to do so may result in material licence breaches, exposing the customer to substantial financial liability or forced renegotiation on unfavourable terms.

Previously enterprise software products were licensed on a named user or concurrent user basis, under which each individual who accesses or uses the system must be covered by a corresponding licence entitlement. The deployment of AI agents or bots that autonomously access, query, or interact with such software raised fundamental question as to whether each such automated actor constitutes a “user” for the purposes of the applicable licence terms. Today, most established providers (e.g. within ERP) operate with license models, which allow deployment of AI agents and bots and those license terms may likely cover the use of Agentic AI also. However, as Agentic AI further develops the use of automated processes, it should be specifically assessed if the use of Agentic AI fall outside the scope of the customer’s existing entitlements. The same considerations apply to indirect use (e.g. where the customer builds its own functionality outside the ERP system, which, however, uses data from the ERP-system. Major vendors have adopted varying and evolving positions on this issue, and customers cannot assume that existing licence grants will accommodate AI-driven access without additional entitlements or fees.

IT projects should be conducted according to a compliance-by-design principle. This means that regulatory and legal requirements are embedded already at the design stage, rather than being validated retrospectively.

This requires that legal and compliance stakeholders are involved early in the project lifecycle to review and challenge the system’s target design (or “blueprint”), ensuring alignment with (i) applicable regulation, such as GDPR, DORA, NIS2, Data Act, Cyber Resilience Act, AI Act and (ii) third-party licence terms (for example, if agentic AI is used).

This early involvement reduces the risk of costly redesign, implementation delays, and regulatory non-compliance, and supports a more robust solution.

Contact information

If you would like to discuss any of the issues raised in this newsletter or require assistance with an upcoming IT project, please do not hesitate to contact our Technology and Commercial team.

Kasper Laustsen
Attorney-at-Law,
Partner

E kasl@plesner.com
D +45 36 94 24 86
M +45 31 33 53 26

Bodil Maria Hald
Attorney-at-Law,
Partner

E bmh@plesner.com
D +45 36 94 23 70
M +45 24 52 40 57

Michael Hur Bertelsen
Attorney-at-Law,
Partner, L.L.M.

E mbe@plesner.com
D +45 36 94 12 54
M +45 29 99 31 08

Plesner Advokatpartnerselskab
Amerika Plads 37
2100 København Ø

33 12 11 33
plesner.com