

Tech/Law

6 Reasons That Tech Projects Go Wrong (And How Your Lawyer Can Help!)

Edition 4 – Not understanding the implications of the commercial model

In an ideal world, every IT contract would be simple and concise, and every customer would have a clear understanding of the commercial terms and how they apply in practice to their project or organisation.

Unfortunately, there is no getting around the fact that IT contracts are often complex, dense texts with a mosaic of defined terms and legalese. This isn't necessarily because lawyers want to make life difficult for anyone – rather it is a sign that Tech Projects and the underlying software products can be complex or have complex pricing and use arrangements.

Understanding the commercial model behind enterprise licencing terms and design solutions is fundamental to negotiating agreements that adequately protect customers when projects go bad.

In this edition, we explore some of the key commercial terms and concepts that customers should be aware of when negotiating and entering into IT contracts for tech projects.

We'll break these down into several sections:



**Tech Projects and
delivery models**



Enterprise licensing



Vendors' fine print



Managing the paperwork



Understanding Tech Projects and delivery models

The commercial fundamentals of a Tech Project and the delivery model behind it are often overlooked in the rush to issue a contract based on an organisation's standard form of technology agreement. How many times have we been told "we need a contract that we can put into the RFT...and oh, we need to issue the RFT tomorrow"? How many times have organisations put a template contract out to market which is not fit for purpose for the project, and does not cover the major risks inherent in the procurement? How many times have we had to renegotiate to include additional risk coverage into contracts where a vendor has already been down selected and priced on the basis of a commercial model that the vendor has put forward?

The challenge: Understanding the fundamentals can be the difference between preparing a contract that incentivises and supports the delivery of a successful project, and one that provides a customer with inadequate remedies and the prospect of having to initiate litigation to recover money paid for an unsuccessful solution.

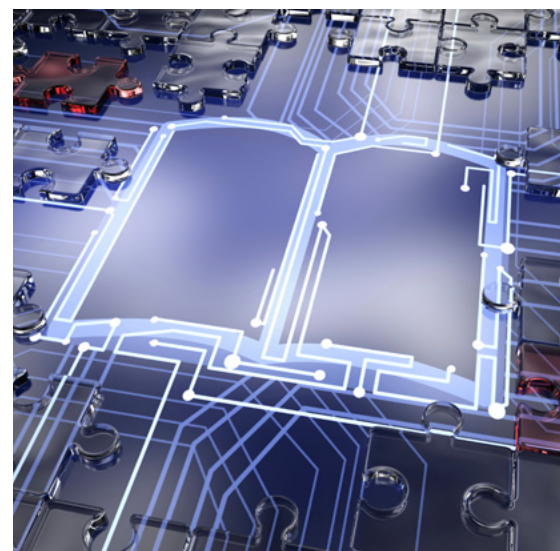
They're simple questions:

- "What are we buying?",
- "How are we buying?",
- "Who is going to be responsible for supplying and integrating it?",
- "How will it be delivered and implemented?",
- "How and when will we pay for it?" and
- "What happens if it doesn't work or is late?"

Some common problems:

- (1) A system integration project that is contracted using a professional services or consultancy contract – leading to a time and materials outcome with indefinite scope and uncapped costs.
- (2) A technology solution procurement that is contracted using a hardware procurement contract coupled with a professional services contract for software development and configuration – the customer can be left with ownership of useless hardware if the software development and configuration is late or unsuccessful.

The solution: Project and procurement teams need to invest time in bringing lawyers up to speed with the key outcomes required. Lawyers need to invest time in understanding the project, asking the right questions and asking the right questions and guiding the project and procurement teams in designing a fit for purpose contract. If successful, the risk allocation, the delivery model and payment structure in the contract will be aligned and clearly defined with successful entry and exit criteria so that the customer can understand and manage successful (or unsuccessful) delivery of the project. Accountabilities for delivery of each stage in the project should be clear and payments should be aligned to successful achievement of stages and delivery of value, rather than simply to deliver cash flow and revenue recognition for the contractor.





Understanding enterprise licensing

In enterprise software contracts, licence metrics define how a customer is allowed to use vendor software, and the way the customer will be charged for that use. Vendors may offer their products and services through several metrics, recognising that there is no one size that fits all.

Here we explain common challenges and solutions for:

1. User licences
2. Processor licences
3. All-you-can-eat licences

(1) User licences

The user-based licensing model has been around for a long time, is generally well understood and continues to hold prominence in today's increasingly SaaS-centred environment. Under this model, the customer effectively purchases a licence for each user in their organisation that needs access to the product or service.

Most user-based licensing models are comprised of several categories of users; for example "named users," "administrator users," "developer users," "employee users" and "professional users". Each has its own set of licence rights and the more extensive the licence rights, the higher the licence fee will be. The user-based model is fairly straightforward and intuitive, however, understanding how the model applies in practice often requires a detailed knowledge of the terms and an organisation's needs.

The challenge: A common issue that arises in this context is where the customer fails to correctly match the correct configuration of user licences to the organisation's usage profile. This can be as a result of a failure to understand the relevant rights that attach to each category of user, but also can also result from the customer's lack of understanding of its own needs or its needs in the future. Without a good understanding of the user categories and how these map onto an organisation's usage profile, customers can end up being under- or over-licensed under this model.

The solution: Customers should ensure that they understand the user categories, the relevant rights that attach to each, and how these categories map onto their organisation. As a rule, the aim is to match each user with the lowest level of licence that still enables them to do their job efficiently. This can be a complex task and may require the assistance of third-party consultants or specialists. Customers should also make sure to factor in projected changes in demand while also continually monitoring its ongoing usage profile.

The challenge: Another common issue that can be traced back to misunderstanding the user-based model concerns the question of what happens when the customer exceeds the number of assigned users or wishes to reconfigure the numbers in each category. For some user-based models, the contract may stipulate that each additional assigned

user above the initial estimate will be subject to increased pricing. Similarly, a customer who wishes to reconfigure the numbers of users under each category may only be allowed to do so subject to the payment of additional fees. Failure to consider how the pricing model reacts to these future events can prove costly. A common pitfall in this regard is where contract makes the issue of additional pricing or reconfiguration 'subject to negotiation'. Where this is case, the customer invariably ends up in a weaker negotiating position as the customer is by this point reliant on the vendor's products and locked-in due to the prohibitive costs involved in migrating to another product or provider.

The solution: The best way to avoid these issues is to undertake proper due diligence to get a clear understanding upfront of organisational needs (with projections of anticipated use) as well as how the pricing mechanisms respond to the customer wanting to increase or reconfigure users. Customers should be wary of making pricing for additional users "subject to parties' negotiation" unless they are confident that they will be in a position to negotiate a favourable outcome.

(2) Processor licences

Processor licences (or CPU or Core licences) allow an unlimited number of users to access and use software that is installed on a processor (CPU or core). Under this model, the customer pays a licence fee calculated per processor being used to run the software. This processor-based commercial model is more commonly used for on-premises server-based applications and is less prevalent now given the growing dominance of SaaS-based models.

The Challenge: Some processor licences may require that the customer obtain a separate processor licence for every processor that is physically capable of running the program in question (even if, in practice, the software will never run on those other processors). Where a customer partitions a multi-processor server through the use of virtualisation software (“soft partitioning”) so that the software can only ever be executed on one processor, the vendor may still consider that a separate processor licence is required.

The solution: Customers should ensure that they have a clear understanding of the specific terms of the agreement when comes to processor licences and how they apply to virtualised environments. The customer should be careful to ensure that their proposed configuration of the server on which the software is installed complies with the terms of the licence.

News Flash: SAP's move to the cloud spells admin dramas for companies

According to the Australian Financial Review, the SAP announcement that it is accelerating its shift to cloud-based computing has major implications, stating that “This will require conducting an audit of everything in a company’s technology stack, as well as an audit of the interdependencies between different software products”.

This means that if companies are looking at their SAP options, it will be even more important than before to understand their rights and obligations under their SAP contracts.

(3) All-you-can-eat licenses

The “all-you-can-eat” licensing model allows a customer unlimited access to vendor’s software or technology for a fixed-fee for a fixed period. It is often used in circumstances where an organisation anticipates rapid growth in the use of the vendor’s software. At the end of the period, there will be a determination of the licenses used in that period, which will set the customer’s licence usage moving forward. Maintenance charges are often set on an agreed basis, which reflects expected usage of the licences over that period.

The Challenge: If the customers do not meet its expected growth targets for usage of the software and fails to get their projected level of use, then they may end up paying more for the software or solution than if they had been on an alternative model. If they use fewer licences than anticipated, they may also pay more in ongoing maintenance fees than that they would have paid had they been on an alternative commercial model.

The solution: Customers should have a clear strategy around how they intend to utilise the software during the term, to maximise the benefits and minimise the costs after the period ends. Customers should also confirm whether support services and other additional features or modules form part of the fixed-fee offering or are subject to additional charges. They need to ensure that they architect their infrastructure to maximise these benefits and track usage on an extremely accurate basis to prepare for post-term negotiations and benefits.



Understanding the implications behind an enterprise vendors' fine print

So far we've discussed some of the core concepts behind commercial models in enterprise licensing contracts. This section focuses on terms and concepts often buried in the fine print. Don't be fooled - their relative obscurity makes them no less problematic if not fully understood.

(1) Indirect use

Commercial models often centre around the concept of the 'user' and or 'use' as a key metric. Many enterprise licences will stipulate that it is not only those who 'directly' use the software that are required to have a licence, but also those who make 'indirect use'. While the concept of 'indirect use' is a feature of many enterprise software licences, it is often not defined or only defined in a vague way. However, its meaning can have profound implications. This issue is highlighted in the Diageo v SAP case study.

Case study - Diageo v SAP.

British beverage company Diageo was operating its own proprietary applications on a Salesforce platform that integrated with an underlying SAP system. Diageo had replaced a manual call centre solution where operators entered customer order information into a Diageo SAP system, with an application where customers could enter their orders directly into the SAP system. The UK High Court was asked to decide whether these users were 'indirectly' using the SAP system. The court found that the terms of the agreement required a named-user licence even if a person only used the SAP software through an intermediary interface or application – meaning not all the users of the proprietary applications were covered by the licences that Diageo had purchased from SAP.

Here we see technological change for customers' benefit having unforeseen consequences for it under the organisation's underlying licence agreement. This shows how fundamental terms like 'indirect use' have wide-reaching implications. Any change in system architecture that may alter the way a customer's organisation intersects with its underlying software platforms needs to be carefully considered against the risk and potential cost of triggering new licensing exposure.

(2) Additional features and modules

Sometimes software packages will come with additional features and modules that may be installed on top of the standard or base configuration. The customer may not want to use all of these additional features, however, vendors may install the fully-featured package onto the customer's system but configured to disable the additional features and modules.

The challenge: If a user in the customer organisation does not understand that certain features are not included in the licence fees and subsequently activates them (whether intentionally or accidentally), then the customer may incur additional licence fees.

The solution: Customers should know what they are getting as part of the contract. If a customer does not anticipate using additional features or modules, they should have appropriate controls against unauthorised activation or insist up-front on installation of the 'right-size' version (with additional features needing to be licensed or installed separately).



(3) Automatic price increases on expiry of term

It is commonplace for enterprise licences to contain a provision which allows for the automatic renewal of the agreement on expiry of the term. This can be a useful feature as it allows the parties to continue the provision and use of the software without having to worry about the renegotiating of a renewal, or worse, using the software on an unlicensed basis.

The challenge: It is common for IT contracts with auto-renewal provisions to link the increases to things like CPI or RPI or other price indices (e.g. the supplier's then standard price list). This can lead to significant increases in price when applied to large projects with significant numbers of users.

The solution: Customers should ensure that they are aware of and understand any automatic changes in price or other characteristics that might take place on expiry of the initial or renewed term.

(4) Back payment of support fees

Support charges for enterprise software can be very significant, nowadays in the range of 20-22% of the base licence fees. Sometimes enterprise licences will allow a customer to selectively terminate support services for their software or solution. Customers may be attracted to the prospect of switching to cheaper support services offerings from a third party supplier and may exercise this right under the agreement.

The challenge: If a customer terminates support and later changes its mind then the vendor may require a back-payment of support fees (to cover the period where the software was unsupported) and a reinstatement fee as a condition of resuming support services. A vendor may also require a back-payment of support fees if the customer later decides to upgrade to the next release of the software. Typically, the fees are equivalent to the amount the customer would have paid for support but for their decision to terminate and switch, although some vendors impose a higher charge for this "privilege".

The solution: Customers should carefully review the terms of their agreement to identify whether they are required to back-pay fees. If there is such a requirement, then the customer should factor in these potential costs in their overall assessment when deciding whether to switch to a third-party support provider.



This process gave our client the necessary information and confidence in its position, to enable it to negotiate a commercially acceptable outcome on its licensing and maintenance position with the vendor.

Keeping on top of the paperwork

Finally, when it comes to understanding the commercial model, it is not only a question of 'what' terms apply, but a question of 'where' those terms are applied. Keeping tabs on the numerous versions of contracts and their various statements, schedules, attachments, addendums and variations can be difficult, but it is fundamental to ensuring that the customer knows their contractual and commercial position at any given point in time.

The challenge: Enterprise software vendors often incorporate boilerplate terms into their agreements that are made accessible on their website and updated from time to time. Vendor contracts may also refer to other policy documents or white papers as a source for substantive obligations or as a guide to interpreting the terms of the agreement. These too may be updated on an ongoing basis. It is also common for the parties to seek variations to their agreement as a project progresses (change being a constant for many tech projects). If a customer makes a commercial decision or enters into an arrangement with an outdated or incomplete picture of their rights and obligations, then this can be costly.

The solution: Customers should have a clear procedure for managing contractual documents and any ancillary documentation. This may involve creating a central repository for copies of all documents with a version control system and assigned custodians whose responsibility it is to ensure that the system is up-to-date and complete.

We hope you find some of this information insightful in understanding contractual considerations relevant to the decisions you make. We would be delighted to discuss any questions you may have. Stay tuned for our next edition!

Cheng Lim & Thomas Dysart
King & Wood Mallesons
October 2020

Recently a client was looking to reduce some software licensing and maintenance charges. The enterprise software vendor was aware that the client had been increasing its automation and customer self-service options that interfaced with the enterprise software. Following discussions on revised licensing and maintenance terms, the vendor sought an audit of our client's usage of its software (as it was entitled to do under the terms of its agreement with our client).

To help our client understand its contractual entitlements, rights and obligations and respond to the audit response, we had to first produce a complete and up to date version of the contract with the vendor. This was a complex exercise. Not only did the contract refer to multiple online documents of the vendor, the parties had executed numerous variation agreements and purchase orders over the years. We had to analyse and aggregate the different types of licences acquired and exchanged by the client over that period. Building an accurate contractual picture was critical to helping the client prepare for its negotiations with the vendor.

We then worked with the client to undertake its own internal audit of its licensing situation (including how the client had implemented its virtualisation strategy). Armed with this information, we could negotiate a detailed protocol for the undertaking of the audit with the vendor - including details of the client personnel who would be made available to the vendor, how the audit would take place, the systems the vendor could have access to, and the information that would be made available to the vendor.

We've got you covered

KWM National Team



Michael Swinson

Partner, Melbourne
TMET, IT & Data
T +61 3 9643 4266
M +61 488 040 000
michael.swinson@au.kwm.com



Bryony Evans

Partner, Sydney
TMET, IT & Data
T +61 2 9296 2565
M +61 428 610 023
bryony.evans@au.kwm.com



Kirsten Bowe

Partner, Brisbane
TMET, IT & Data
T +61 7 3244 8206
M +61 409 460 861
kirsten.bowe@au.kwm.com



Cheng Lim

Partner, Melbourne
TMET, IT & Data
T +61 3 9643 4193
M +61 419 357 172
cheng.lim@au.kwm.com



Patrick Gunning

Partner, Sydney
Tech, Data & IP
T +61 2 9296 2170
M +61 418 297 018
patrick.gunning@au.kwm.com



Nicole Heller

Partner, Sydney
TMET, IT & Data
T +61 2 9296 2347
M +61 417 213 334
nicole.heller@au.kwm.com



Renae Lattey

Managing Partner, Clients and
Mergers & Acquisitions
T +61 3 9643 4065
M +61 417 214 795
renae.lattey@au.kwm.com



Annabel Griffin

Partner, Canberra
IT & Data
T +61 2 6217 6075
M +61 408 847 519
annabel.griffin@au.kwm.com



Rachael Lewis

Partner in Charge, Canberra
TMET, IT & Data
T +61 2 6217 6074
M +61 448 056 645
rachael.lewis@au.kwm.com



John Swinson

Partner, Brisbane
Tech, Data & IP
T +61 7 3244 8050
M +61 408 220 513
john.swinson@au.kwm.com



Kate Creighton-Selvay

Partner, Melbourne
TMET, IT & Data
T +61 3 9643 4071
M +61 405 993 122
kate.creighton-selvay@au.kwm.com

www.kwm.com

Asia Pacific | Europe | North America | Middle East

King & Wood Mallesons refers to the network of firms which are members of the King & Wood Mallesons network.
See kwm.com for more information.

© 2020 King & Wood Mallesons