SUBMISSION BY

to

INLAND REVENUE DEPARTMENT

on

INCOME TAX TREATMENT OF SOFTWARE DEVELOPMENT EXPENDITURE AND SAAS CUSTOMISATION AND CONFIGURATION COSTS

30 January 2026

CONTACT:
Graeme Muller
Chief Executive
NZTech
E | Graeme.muller@nztech.org.nz **M |** +64 21 0252 0767

SUBMISSION ON INCOME TAX TREATMENT OF SOFTWARE DEVELOPMENT EXPENDITURE AND SAAS CUSTOMISATION AND CONFIGURATION COSTS

30 January 2026

INTRODUCTION

TechNZ thanks Inland Revenue for the opportunity to contribute to this consultation by responding to its issues paper.

ABOUT TECH NEW ZEALAND

TechNZ is a member-funded, not-for-profit organisation representing more than 2,500 members who together employ 10 percent of the New Zealand workforce, comprising startups, local tech firms, multinationals, education providers, financial institutions, major corporations, network providers, hi-tech manufacturers, and government agencies that work closely with the tech ecosystem.

The tech sector is a significant and growing part of the New Zealand economy, employing 121,000 people and contributing around $22b in GDP. It is also one of the fastest-growing export sectors – New Zealand's 3rd largest – with export receipts of $10.7b in 2023. Software exports, for example, are growing at more than 20% p.a.

We note that "tech" includes physically manufactured products with a significant digital/knowledge-intensive component (e.g. Rakon, Tait Communications, F&P Healthcare), as well as weightless "digital" exports such as software, AI or gaming (e.g. Datacom, Xero, Orion Health, RocketWerkz).

RESPONSE TO QUESTIONS RAISED IN THE ISSUES PAPER

Chapter 2 – Software development expenditure

Do existing rules provide appropriate basis for deductibility or depreciation of software development expenditure?

Q1. What is your current approach to determining the tax treatment of software development expenditure?

There are really only two accounting treatments of qualifying software development expenditure: it's either a cost of doing business (operating expenditure), or an investment (capital expenditure). If it is capital expenditure then we consider the use of grants, tax treatment, and building enterprise value. As accounting treatment supports all management activities, not just tax, this is the foundation. Where tax rules do not align with the nature of the expenditure, a burden is created to rework the accounting records to match the tax perspective for filings. Often it’s worth tweaking the accounting treatment to reduce the tax compliance burden, even if it negatively impacts other decision support systems.

Authoring robust, defensible accounting principles are helpful where tax rules are subjective, unclear, or ambiguous. As part of that documentation we try to explain where tax rules are difficult to apply and our assumptions/efforts to ensure our accounting principles meet compliance requirements. We also obtain external expert advice where appropriate. Obtaining a binding ruling is an expensive, time-consuming exercise that has always resulted in the same, conservative interpretation we could have made without the ruling. Unless there is a material risk to mitigate, we do not seek binding rulings.

Q2. Do you consider that the approaches described above (that is, the trading stock, depreciation, and R&D treatments) are likely to result in the correct outcomes for tax purposes? What issues arise with applying a given approach?

These approaches are technically flawed or incomplete, but ironically reach appropriate outcomes for tax purposes in many situations.

A big issue is that software development is a generic term, used broadly to describe vastly different activities and artefacts. Fixed rules for loosely defined activities, or loose rules to accommodate the variety of activities, are a primary source of uncertainty and complexity. We prefer to avoid the term “software development” as we have better, more precise language these days to categorize activities.

Another big issue is the linear thinking that software is developed then sold. Software is the codifying of computer instructions, in the same way data is codifying for storing observations.

The *intellectual property rights* are sold, the knowledge assets themselves are copied or shared. Reference to developing software products is technically incorrect, as we develop software assets that have no tangible value. A product is created when an intellectual property *ownership* right is issued. A service is created when an intellectual property *use* right is issued.

Q3. We understand that the timing of deductions may be a particular concern. If so, how significant for you is the difference in the timing of deductions allowed under each approach?

Where SaaS offerings are a significant source of income, cashflow is complex to manage. A substantial portion of expenditure is disconnected from income. Circumstances can be variable:

- Expenditure can be lumpy whereas income ramps up over time.
- Some situations are the opposite, where income is front-end loaded before expenses are incurred.
- New income streams can be formed with new product or service offerings leveraging existing software.
- Software may be offered for free while value-added services are wrapped around them.

Timing of tax payments and deductions have a material impact on cash flow.

Two things really matter: (1) Inland Revenue holding cash that could be returned to the business; (2) uncertainty over timing or entitlements.

A hypothetical example: a business is unsure if Inland Revenue will agree with the scope of expenses it wants to claim as R&D. Consequently, it decides to pay provisional tax at a level that will avoid penalties/use of money if IRD doesn't agree.

Two outcomes are possible here: (a) the business's R&D tax deduction claim is reduced/rejected, in which case it is disappointed that IRD has taken such a narrow approach; or (b) IRD accepts the R&D deduction claim and the business is e disappointed that IRD held cash that it could have used to grow its operations.

It's not actually about the timing. It's about not hindering business efforts to maximise use of cash, and not making it stressful. The closer deductions/payments are to the business activity, the more they fuel that activity. Realising a deduction or payment benefit months or a year later is fine for rewarding behaviour, but the longer the delay, the less incentivising it becomes.

Q4. When a software product is developed to be sold rather than licensed, is it still appropriate to apply the trading stock rules? Should the tax principles be standardised across all forms of software development?

The trading stock approach worked fine when software was distributed mostly through physical media, even though technically it was the wrong approach, because the software was not being “sold”. But it worked because the use rights licenses sold were attached to the physical media. The equivalent today is issuing keys to unlock user rights to software that is downloaded, or establishing an account to access software that is hosted.

Using trading stock rules for software is like treating road rules for cars as if they’re horses with wheels. It can work but it will be ugly, and compliance will be odd.

Whenever a business retains ownership of the intellectual properties it creates, it issues a license agreement when a customer buys rights to use that intellectual property. Regardless of whether it's packaged as a product or service, customers are purchasing use rights through a license.

Software development is also really problematic as a term. Data and art make up a huge portion of the shareable digital assets, but they are technically not software as inferred in tax discussions. There is no functionality, but it is expensive to accumulate data and create art.

Art is creative expression, and determines important product aspects such as user experience. The design knowledge of what “art” works and what doesn’t is the intellectual property that is researched and developed, and is an essential element to the practical application of technology to a problem.

Data is codifying of factual knowledge. Just like art, it is the knowledge of how to codify the facts, rather than the data itself, that is the intellectual property.

Today we create digital products and services, and software development is one of the activities. Standardising tax principles across the offering of digital products and services, and distinguishing between development, adoption, and utilization would go a long way to simplifying efforts to standardize principles.

Q5. Are the asset recognition criteria under NZ IAS 38 suitable for identifying when a software development project has generated a recognisable asset, in the context of determining appropriate tax treatment?

Yes and no. If a software development project produces a digital asset with future economic benefit, then it has produced something that requires future maintenance and adaptation. It is an intellectual property with ownership responsibilities. This both meets the recognition criteria and conflicts with some of the assumptions about intangible assets described in NZ IAS 38

paragraph 20. Software intellectual properties are functional, whereas other intangible assets are not.

If a software “engineering” project implements software obtained from a third party, then it incorporates the intellectual property owned by the third party into its internal systems. Configuration and integration tasks are transitional expenses to “hook up” the new third-party component. This is equivalent to replacing the fuel pump in a car – a new component is added to the fuel system, and the cost of integrating it is an expense.

To complicate matters, a long-term contractual commitment to license the use of third-party software may be considered an intangible asset because of the future benefit. Typically these days there is also an associated liability, so that costs are incurred at the same time as benefit is realised. In practice, whatever value is attached to recognising the third-party license as an intangible asset with future benefit would be matched with recognising any future liability to pay for its use. If customers negotiate payment terms that do not align with utilization, they can create a prepayment or deferred payment scenario. Existing tax rules cover these scenarios, and the fact that software is involved is irrelevant.

Q6. Do any aspects of applying NZ IAS 38 cause escalated compliance costs in the context of claiming tax deductions under section DB 34? Do the compliance costs affect your decision to use section DB 34 to calculate appropriate deductions?

There are increased administration costs anytime there are additional regulatory compliance requirements added, even if they substantially overlap/align with others. Just within tax we have the R&D Tax Incentive, R&D Tax Credit, IAS 38 and Section DB34 to reconcile when deciding on appropriate accounting. We also need to factor in compliance with any grants or other financial assistance we’ve received, as well as meeting the financial objectives agreed with the Board of Directors.

There is a view that DB34 uses overengineered language, making it torturous to reconcile with IAS-38. After all is mapped out, we could spend considerable sums of money untangling the knot only to find we just needed to apply common sense. At the same time, some smaller businesses will look at DB34 and avoid it because it is expensive to interpret, so IAS-38 is useful in its own right.

Q7. Is the nature of software R&D sufficiently distinct to justify a case for simplifying the approach for software development by addressing software R&D in a standalone provision instead of section DB 34?

Yes, it would be helpful to have a dedicated section on development expenditure of software, as well as other digital assets such as data or art. Software is unique as a technology, as it is the only functional intangible asset class. It is a substantial area of investment, expense, and development for businesses.

However, software development is a difficult term to work with as it encapsulates too many activities. Building digital assets, offering digital products and services, utilising digital services, implementing digital products, etc are all definable activities or identifiable assets. Some sharper terminology would dramatically simplify all the “ifs, ands, ors, and buts”.

Non-IFRS taxpayers and section DB 34

Q8. Are the compliance costs for non-IFRS taxpayers of using NZ IAS 38-based requirements in section DB 34 prohibitive to the use of that section? If so, what alternative approaches would result in appropriate outcomes for both IFRS and non-IFRS taxpayers?

Many businesses adopt IFRS as guidance, if they are not IFRS taxpayers. It is often used for awarding grants, or meeting debt obligations. DB34 can be made more useful if its wording is revised to remove the overly technical language.

Asymmetric results possible under section DB 34

Q9. Do you agree that an equitable treatment would ensure that assets supported by deductions made under section DB 34 are not treated differently from other assets that, if sold for greater than their tax book value, are subject to normal depreciation recovery? What justification might there be for a differential treatment?

Digital assets are expensive to build, but they have no inherent value. It is the intellectual property ownership rights that are commercialized that have market value. Digital assets are not sold, they are shared or copied. Intellectual Property ownership and use rights are sold. This makes software fundamentally different to other intangible assets, which have no utility. Software IP is valued for its utility, whereas the software itself has value based on its uniqueness and investment. Over time that uniqueness erodes, justifying the depreciation if expenditure is capitalised.

Developed software IP depreciates rapidly as it goes from being unique in the world, to novel, to widely copied. The commercial value of IP licensing to customers is independent of the investment in the underlying digital asset. This is different to other market-valued intangible assets.

Where software has been developed for internal use, it becomes part of the internal systems for a business. Consequently the software has no distinct value itself, only as a component of a system that has the intangible value. It is unreasonable to attribute the value of a system to its components, so depreciation recovery on the software development components separate from the system is complicated.

Chapter 3 − Software as a service

Income tax treatment of SaaS C&C costs

Q10. Do you agree that the approaches to the income tax treatment of SaaS C&C costs described above lead to the correct outcome for tax purposes? If not, what do you consider are the problems?

This is an incomplete structure for tax treatment of SaaS implementation and the utilisation costs incurred by end user businesses. Most of the cost of integrating a third party SaaS into an internal system is configuring other components to work with the new SaaS component, migrating data from the old software being replaced, and building any new functionality to glue it all together. Configuration and customisation have a much broader definition within business than the narrow scope defined by IFRIC. There is considerable debate during sales negotiations over whether implementation costs can be capitalised or expensed. Aside from enterprise pillar SaaS applications such as ERP or CRM, most of the implementation costs are outside the SaaS product.

The most important sentence is in Section 3.16: "*This enduring benefit is linked to the business structure and is not part of the day-to-day operations*." In all SaaS deals with substantial implementation costs, the driving factor in determining whether costs were expensed or capitalised was the business structure. If the business was capital intensive with enduring value coming from their systems, then capitalising the implementation costs was usually the appropriate accounting treatment. An operationally intensive business with low capital investment would typically incur expenses to use the SaaS software, rather than invest in an enduring integrated system. The business would want to expense the implementation costs as there's no enduring value.

Section 3.18 is too simple in its interpretation of SaaS use rights. Use rights are typically made available and consumed monthly. That is not a depreciable intangible property, it is the consumption of a service. If the license agreement is treated as a depreciable intangible property, then the financial commitment in the license agreement needs to be treated as a liability of equal value. Over the life of the license agreement, the property value amortizes at the same rate as the liability is paid down.

Narrowing the focus on the SaaS product instead of the system integration over-complicates matters and does not address the bulk of the expenditure made when adopting a new SaaS product.

Q11. If innovation were part of the test for section DB 34, do you consider SaaS C&C costs would likely meet this criterion? If SaaS C&C costs cannot meet the criterion of innovation, is it justifiable that these costs meet the definition of R&D for tax purposes?

The New Zealand Government has been inconsistent in its definition of innovation, making any incorporation in regulations complex for software investments. “Configuration” is setting the properties of software to meet user/operator requirements. By definition, this is utilising existing knowledge rather than creating new knowledge, and so would not meet any definition of innovation. “Customisation” is creating new code that applies digital technology to a specific requirement, because the existing application does not meet the needs. This does meet some definitions of innovation.

Where a SaaS application is implemented standalone, then the IFRIC guidance on SaaS C&C is appropriate. If the SaaS application is being integrated into internal systems, then there is innovation at the system level, and developing new knowledge around the correct configurations to make the entire system operate is a form of innovation.

Innovation is already handled in the tax system through R&D Tax Incentives and R&D Tax Credits. DB34 deals with R&D expenditure regardless of innovation. If someone decides to reinvent something for commercial reasons, it is a legitimate R&D business expense but it is not innovative. “I can do it cheaper myself” is a valid option for businesses. If innovation was part of the test for DB34, it would exclude legitimate R&D business expenses and overlap other tax programs targeted at innovation.

Q12. Is it the correct outcome that the tax treatment of SaaS C&C costs varies depending on whether a taxpayer carries out the SaaS C&C work in-house or through the SaaS provider or a third party? If not, what problems arise from this outcome?

This is a difficult way to frame the question, as SaaS C&C is such a narrow scope for client-side software R&D expenditure. Regardless, who does the work and how they are contracted/employed should not be the criteria for determining tax treatment. What matters is the business structure the SaaS application is being incorporated into and the commercial activity that pays for it. If a business’s contract says it’s buying IP ownership or use rights, does it matter from an accounting/tax perspective if the charging mechanism is based on time, labour, or deliverables? If the business does it itself, it's outside the contract for the SaaS application and gets treated like any other software development effort the business might undertake.

Existing tests can already produce appropriate tax treatment. What is the problem this is trying to solve?

By way of context, customisation is becoming yesterday’s problem. SAP has provided a worldwide lesson in the massive strategic risk associated with SaaS code customisations. All custom modules require rework across design, development and deployment before being released into production on the new underlying database. Most B2B SaaS applications now

provide programming interfaces (API's), support automation tools, or other integration features, and discourage client-specific code customisations of the SaaS application.

Q13. Do you consider the compliance cost of applying section DB 34 to SaaS C&C costs a significant impediment to applying the section? If so, is your main concern rules that minimise compliance costs or are there other concerns that we need to consider?

Absolutely. DB34 would be difficult to apply to SaaS C&C costs, because the client R&D costs are associated with resolving system uncertainty rather than developing a better functional fit within the SaaS application. By taking a system-integration perspective we avoid all the SaaS C&C specific concerns and have easier, stronger compliance where DB34 is relevant.

The UK government has produced some good guidance around systems uncertainty and its relation to R&D. This was in relation to its R&D tax incentive program, which New Zealand's program was largely based on. The UK government had to update its program to clarify handling of systems uncertainty specifically for software R&D expenditure. This was required because its R&D policy is based on the Frascati manual, which focuses on turning scientific discovery into developed applications. The Oslo manual is another useful policy guide as it approaches R&D from an innovation perspective and avoids the "practical application" challenges that Frascati creates.

Q14. Do you consider applying the depreciation rules a suitable alternative to a deduction under section DB 34? If not, what problems would arise from this approach?

R&D is an activity with real costs. It is not an asset that depreciates. R&D resolves uncertainty and creates new knowledge whereas depreciation recognises value deterioration. When building software the value of the IP asset created has no correlation to the R&D effort to create it. The value of a software IP asset is in its utility.

For internal use, the software becomes a component in a system and the value is in the overall system. Again, the software IP asset value is in its utility, not the R&D effort to create it.

If R&D activity is deducted as an expense, then expense rules apply. If R&D activity is deducted as depreciation, then depreciation rules apply.

An example: If someone spent money one year building a car, then sold that car the next year for the same value, they would have broken even. If it was treated as depreciation, then from an accounting perspective they've overestimated the depreciation and should reverse the depreciation when they sell the car. Even though they actually spent real money and made no real profit, they would have to reverse 100% of the costs and recognise 100% of the revenue as taxable profit.

Q15. Does this issue cause you any practical issues? For instance, does inconsistency in the period over which SaaS arrangements of different legal lives are depreciated create higher compliance costs for you?

The only SaaS arrangements this is relevant to would be poorly constructed contracts for major SaaS implementations at large enterprises/government agencies. Even the most creative deal making will break down the cost components into what is prepaid, what is owing, and who owns what, so it's easy to budget for and buy. If R&D is required as part of the deal, it is declared along with ownership and deliverables. These terms determine accounting and tax treatment, making much of this topic irrelevant.

To be absolutely clear here, nearly all SaaS arrangements are for Licensed Use rights to access services hosted by the vendor. The arrangements do not establish any ownership of intellectual property by the client. SaaS stands for “Software as a Service”. The intangible asset is the contract and use rights agreement, not the actual software. Consequently there is no reason to treat SaaS contractual arrangements differently to other contractual arrangements. If use rights are prepaid, there’s an asset. If use rights payment is deferred, there’s a liability. If the vendor commits to performing R&D for IP they will own, then they will incur costs which they recover from charging for use rights. If the client wants to fund R&D that they will own, then it's outside the IP licensing agreement for IP the SaaS vendor owns.

This topic does not describe the problem. Yes, complex contractual arrangements make compliance more difficult, and that applies to everything. How does SaaS make it any different? Generally we strive to make SaaS contractual arrangements align with normal business practices as much as possible to make it easy for clients.

Q16. If you experience negative outcomes, do you think this issue requires a solution? Or is this issue not significant enough for you to require remediation?

We are struggling to see the issue here. If a SaaS arrangement establishes any form of asset for a client, it will typically also create a liability that matches the life of the asset. Most arrangements strive to align costs incurred with realised value from usage, so any depreciation is offset by the reduction in liability. If there are any up-font costs that are recovered across the life of the license agreement, then another asset/liability pair are created. There is nothing special about SaaS that would require special accounting or tax treatment.

Black hole expenditure on some abandoned SaaS arrangements

Q17. Do you agree that there is a risk of black hole expenditure in relation to abandoned SaaS arrangements that have no fixed legal life or a legal life of greater than four years?

Paragraph 3.38 highlights the issue perfectly - SaaS arrangements do not assign copyright ownership to the client, so there is no intangible property of the SaaS software owned by the

client. It is technically incorrect to say a SaaS arrangement is an intangible property - there is no ownership. SaaS arrangements have a license agreement that grants use rights to the client over a period of time. The agreement may have value and might be recognised as an intangible asset if it has monetary value, just like any long-term service agreement. If a contractual arrangement is abandoned and there is residual value in the contract, that value can be written off. Treating SaaS arrangements as if the customer has acquired an intangible property right to the software is a root cause of many of the concerns and complexities described in this document.

Everything else is contractual, so long as a business manages SaaS arrangements the same as it manages other service contracts, all the SaaS intangible property and depreciation complexity can be avoided. Expenditure is either expensed or capitalised based on what it's for. If it's capitalised then abandoned, it's recognised in the write-off. Black holes are avoided.

Q18. If you consider this risk to be a potential issue, do you consider that it requires remediation?

If Inland Revenue requires businesses to recognise contracts for services as ownership of assets for SaaS arrangements, then yes, this will create issues such as black-holed costs. The simple remediation would be to recognise Software as a Service contracts as contracts for services, and treat associated activities such as R&D as activities rather than assets.

CONCLUSION

TechNZ thanks Inland Revenue for the opportunity to make this submission. We would like to make sure our members play an active role in the development of any changes relating to software tax treatment, and can easily facilitate further engagement when needed.

Yours sincerely,

Graeme Muller
Chief Executive
TechNZ
E | Graeme.muller@nztech.org.nz **M |** +64 21 0252 0767

Roger Ford
Subject Matter Expert
TechNZ