NZGOAL Software Extension

Version 1
[ ] 2016

New Zealand Government

ISBN [ ] (PDF)
ISBN [ ] (HTML)

Contents

Introduction

Purpose

1 Government agencies invest significantly in software development and often own the copyright in the software that they develop or that is developed for them. Public sharing and open source licensing of this government-owned software has the potential to:

(a) save agencies time and money, resulting in a more efficient use of scarce resources;

(b) encouarge open innovation on the part of both the public and private sectors;

(c) contribute to economic growth, primarily through the private sector being able to leverage and support government investment in the software it openly releases for re-use;

(d) contribute to the formation of trusted communities of users whose public and private sector members have common or similar goals or interests;

(e) result in continuous and ongoing maintenance of the released software code through these communities of users in a way that may not be achievable by a single agency alone;

(f) enable agencies to better align their operational and strategic activities with relevant aspects of the Government ICT Strategy and Action Plan to 2017 (as revised in 2015); and

(g) in some cases, foster transparency – for those who can read software code – as to the methods or algorithms used for the creation or delivery of public data and services, thereby enabling critical analysis and potentially the provision of improvements back to the releasing agency

2 This NZGOAL Software Extension (**NZGOAL-SE**) provides agencies with a means of realising this potential. It:

(a) explains the legal and policy context that is relevant to agencies' open source licensing of software;

(b) sets out a series of policy principles to guide agencies in their open sharing of software code;

(c) advocates the use of particular open source software licences for this purpose; and

(d) sets out a review and release process to guide agencies through the review of the software they propose to release for re-use, the purpose of which is to help agencies make decisions that are legally robust and practically useful.

Open source software and open source software licences

3 For the purposes of NZGOAL-SE, open source software (**OSS**) is software that is released on terms that make it freely available for others to use, modify and distribute, for either non-commercial or commercial purposes, as long as they comply with the applicable licensing conditions. The licensing conditions that users of the software need to comply with depend on the form of open source software licence applied to the software.

4 There are two main forms of open source software licence:

(a) **fully permissive licences** (also known as academic licences) that confer broad freedoms and minimal obligations on those who wish to use, adapt and distribute the software (e.g., the MIT licence, the BSD licence and the Apache 2.0 licence); and

(b) **sharelike licences** (also known as copyleft licences) that confer similar freedoms but require those who adapt the licensed software to license their adaptations with the same licence if they distribute them (e.g., the GNU General Public License, or GPL for short).

Approach and scope

5 NZGOAL-SE is an extension of, and is modelled in part on the approach to, the open licensing of government copyright works set out in the New Zealand Government Open Access and Licensing framework (**NZGOAL**). NZGOAL-SE is a self-contained extension or framework, rather than being part of the original NZGOAL framework, because certain considerations are different to those relating to open information and data. Incorporating both frameworks under a single roof would unduly complicate matters for agencies and others interested in the frameworks.

6 NZGOAL-SE is not concerned with the proprietary versus open source debate or with the considerations that agencies may wish to take into account when using existing open source software for their own internal purposes.[1] Its sole focus is on the public release and open source licensing by agencies of software they own or are authorised to release and license.

Additional guidance notes

7 Additional guidance notes may be released over time which:

(a) explore, in greater detail, some of the issues addressed or raised in NZGOAL-SE; and

(b) address operational or technical issues which arise in practice.

Legal and policy context

Copyright law

8 Key aspects of copyright law relevant to open licensing generally are set out in the NZGOAL Copyright Guide.[2] Additional aspects relating specifically to software include the following:

(a) **copyright exists in original software**: software is a form of “literary work” and original literary works are a category of works in which copyright exists (section 14(1) of the Copyright Act 1994 and the section 2(1) definition of “literary work”);

(b) **infringement**: as such, unless entitled to do so by a copyright licence or statutory provision, a person infringes copyright in software that that person does not own when he or she does any of a number of “restricted acts”, the most common of which is copying the work or a substantial part of it or adapting the work;

(c) **adaptations**: an "adaptation", in relation to a literary work that is a computer program (i.e., software), includes "a version of the program in which it is converted into or out of a computer language or code or into a different computer language or code, otherwise than incidentally in the course of running the program" (section 2(1));

(d) **first owner of copyright**:

(i) **employee-employer relationship**: when employees create software for their employer in the course of employment, the employer is the first owner of copyright in the software unless the employment contract states otherwise. This is the case where the employer is the Crown (section 26) and where the employer is another person, agency or entity (section 21(2));

(ii) **service provider-customer relationship**: when a service provider creates software for a customer under a contract for services, the customer is the first owner of copyright in the software unless the contract for services states otherwise. Again, this is the case where the customer is the Crown (section 26) and where the customer is another person, agency or entity (section 21(3)).

---

1 On these topics, see the Australian Government’s *A Guide to Open Source Software for Australian Government Agencies* (version 2.0, June 2011) at http://www.finance.gov.au/files/2012/04/AGuidetoOpenSourceSoftware.pdf

2 See https://www.ict.govt.nz/guidance-and-resources/open-government/new-zealand-government-open-access-and-licensing-nzgoal-framework/ss-and-licensing-nzgoal-framework-nzgoal-copyright-guide/

Guidelines for Treatment of Intellectual Property Rights in ICT Contracts

9 These Guidelines[3] are relevant where an agency procures the development of software from a service provider. They state that only in limited circumstances should the government own and exploit new intellectual property rights (**IPR**) created under an ICT contract. The default position is that the supplier should own the new IPR, with licences being granted to the customer agency and other State Services agencies.

10 This position could create an obstacle to an agency licensing developed software on open source terms because the agency can only do so if it has the requisite rights to do so. If it doesn't own the copyright, it doesn't have the requisite rights unless expressly authorised by the copyright owner.

11 The Guidelines recognise, however, that there may be situations where an agency needs to own the IPR. One of those situations is where the agency "intends to allow free use of the IPR on open source terms". Agencies would, therefore, be acting consistently with these Guidelines if they were to require the development contract to confer ownership of copyright in the software on the agency, on the basis that they wish to license the software on open source terms.[4]

Government Rules of Sourcing

12 Who is to own the copyright in new software is a significant issue that needs to be considered at the time of procuring the software development services, a point that is made clear in Rule 61(1) (Intellectual Property) of the Government Rules of Sourcing.[5] That Rule states:

> "If an agency's procurement of goods, services or works involves the supplier creating new Intellectual Property, the agency **should** set out, in its Notice of Procurement, its intentions regarding ownership, licensing, and future commercialisation of that Intellectual Property."

---

3 The Guidelines are available at https://www.ict.govt.nz/assets/Uploads/Documents/ipr-guidelines-2008.pdf

4 Where the agency owns the copyright, it could still grant a non-exclusive custom licence to the service provider, before or at the same time as releasing the software on open source terms, that enables the service provider to use the software for other purposes. For example, the service provider may wish to be able to use the software it has developed for other projects before the agency releases it on open source terms.

5 The Government Rules of Sourcing are available at http://www.business.govt.nz/procurement/for-agencies/key-guidance-for-agencies/the-new-government-rules-of-sourcing

NZGOAL-SE Policy Principles

Introduction

13 Government agencies are strongly encouraged to apply the following Policy Principles in relation to publicly releasing and licensing their software on open source terms. The Principles should be read in their entirety but here's a summary :

Policy Principle	Summary
Public release and open licensing of agency software with MIT licence as default	License software with a fully permissive OSS licence. The MIT licence is the recommended licence.
Ensuring copyright ownership or right to sub-license	Make sure your agency has the required copyright-related rights to license the software.
Exceptions	The default MIT licensing does not apply where an exception applies.
Adaptations	Be careful when considering whether you're adapting pre-existing software.
Alternative OSS licensing	Where the default MIT licensing does not apply due to an upstream OSS licensing obligation (e.g., under the GPL), license with upstream licence. Where the agency has a compelling reason for sharealike licensing, license with the GPL (or a GPL-compatible licence).
Potential stifling effect of sharealike licensing	Beware that sharealike licensing (when not required) may appear paternalistic and/or be commercially inhibiting.
Security code review	Consider whether a security code review is required to identify sensitive elements that may need to be removed.
No additional controls or discrimination	Do not seek to impose requirements that are inconsistent with the freedoms in the chosen OSS licence.
No charging	Do not charge people for access to OSS-licensed software.
Updating open source licensed software	If you spot a bug with software you've licensed, consider informing users and, if you fix the bug, release the updated files. Take appropriate action if there's a security risk.
Code forking	If you take and modify open source licensed software, you should contribute the modified software back to the open source community.
Obtaining rights when procuring or commissioning the development of software	If you commission the development of software that you want to OSS-license, make sure you obtain the rights to license it on OSS terms.
Act fairly towards developers when drafting IP warranties and indemnities	If you commission the development of software that uses pre-existing third party OSS-licensed software and/or that you may OSS-license, act fairly in relation to IP warranties and indemnities.
Liability	Replicate all disclaimers and exclusions contained in the relevant OSS licence when you release software under that licence.
Review and Release Process	Follow the NZGOAL-SE Review and Release Process.

Each Policy Principle is set out below.

Public release and open licensing of agency software with MIT licence as default

14 If government agencies have the required copyright-related rights to do so, they should make their software, that is or may be of interest or use to people, available for re-use under a fully permissive open source software licence, unless an exception in paragraph 18 applies (the **Permissive Licensing Principle**).

15 For the purposes of NZGOAL-SE, the recommended fully permissive open source software licence is the **MIT licence**.[6] This licence is a simple licence that grants people almost unlimited freedom with the software as long as they retain the copyright and licence notice and the disclaimer of warranties and liability. The full text of the MIT licence can be found on the website of the Open Source Initiative.[7]

Ensuring copyright ownership or right to sub-license

16 Paragraph 14 makes it clear that, for the Permissive Licensing Principle to apply, an agency must have the required copyright-related rights to license the software.

17 Agencies should only license software for re-use by others (whether under the MIT licence or any other open source software licence) where:

(a) they own or can obtain an assignment of all copyright in the software and have not exclusively licensed it to a third party; or

(b) to the extent they do not own the copyright, they have or can obtain permission from the copyright owner(s) to do so

(the **Rights Clearance Principle**). It can be important, in this context, to check that developers have not copied tracts of code from elsewhere.

Exceptions

18 The Permissive Licensing Principle does not apply where:

(a) **civil wrongs**: open source licensing of the software would constitute a breach of contract, breach of confidence, breach of privacy, disclosure of a trade secret or other actionable wrong;

(b) **commercial interests**: open source licensing of the software would be contrary to an agency's own or the Government's legitimate commercial interests (note, however, that most government agencies are not in the business of commercialising developed software);

(c) **security or privacy risk**: release of the software on open source terms would create an unacceptable security risk (whether to an agency, organisation or individuals) or an unacceptable privacy-related risk; or

(d) **sharealike required**: the agency is either required or has compelling reason to license the software with a sharealike (copyleft) licence such as the GPL.[8] For example:

---

6 The MIT licence is considered preferable to the BSD 3-Clause licence (another commonly-used permissive open source software licence) because the MIT licence includes a clearer grant of rights and expressly includes the right to sub-license (the BSD licence does not). Whilst many interpret a right to sub-license as being implicit in the BSD licence, the absence of an express reference to it in the BSD licence could produce uncertainty for users, most notably users who wish to incorporate government-produced code in a work licensed under the GPL. The Free Software Foundation considers the BSD licence to be compatible with the GPL but that must depend on a particular interpretation of the BSD licence wording (see generally A Sinclair "License Profile: BSD" IFOSS Law Review, 2(1) pp. 1-6). The MIT licence doesn't contain the BSD's 'no endorsement' clause but, in most cases, the law would prevent claims of endorsement without permission anyway. Another common permissive licence, the Apache License 2.0, was not chosen because it is more complex. The Free Software Foundation considers it preferable to the BSD and MIT licences as it deals with patent licensing and prevents 'patent treachery'. However, in New Zealand the Patents Act 2013 excludes computer programs "as such" from patentable subject matter (and, in any event, historically government agencies have not generally been in the business of applying for software patents). NZGOAL-SE could have recommended its own bespoke permissive licence instead of the MIT licence but that would have contributed to further licence proliferation and exposed developers to a licence they're not familiar with.

7 See https://opensource.org/licenses/MIT

8 This form of licence requires those who make adaptations or derivatives of the software to license those adaptations or derivatives under the same licence when they distribute or convey their adaptations or derivatives

(i) the agency may be incorporating or adapting pre-existing code that is licensed under a sharealike open source software licence and therefore be required, by the licence's reciprocity obligation, to licence the software under the same licence upon distribution; or

(ii) the agency (either alone or with agencies and users) may have a genuine need for developments to the software to be shared with the developer community (when considering this situation, agencies should take into account the "Potential stifling effect of sharealike licensing" principle below).

Adaptations

19 Difficult legal questions can arise as to whether a given piece of developed software, that in some way leverages or interacts with other software, is an adaptation or derivative of that software. If:

(a) an agency's developed software does leverage or interact with other software; and

(b) that other software is third party proprietary software or software owned by a third party that has been released under a sharealike open source software licence like the GPL,

the agency should be cautious about concluding that its developed software is *not* an adaptation or derivative of the other software. The need for caution arises because if the agency concludes mistakenly that the developed software is not an adaptation, release of the developed software under the MIT licence could result in the agency breaching third party rights and expose it to the risk of complaint or legal action. End users would also be at risk of breaching third party rights and be exposed to the risk of complaint or legal action. If an agency is in any doubt on this issue, it should seek specialist advice (which could be both technical and legal) before releasing the software for re-use.

Alternative OSS licensing

20 Where the Permissive Licensing Principle does not apply because:

(a) the software is an adaptation or derivative of open source software that is licensed under a sharealike OSS licence, the agency should release the software and license it under the same sharealike OSS licence (in most cases the licence will be version 2 or 3 of the GPL); or

(b) the agency has compelling reason for licensing the software with a sharealike licence, the agency should release the software under the GPL.

21 Whilst versions 2 and 3 of the GPL differ in certain respects, in essence the GPL allows people to copy and distribute the software, to charge a fee for transferring it or providing warranty protection, and to modify the software and distribute resulting derivative works. But, if a person distributes his or her derivative work, that person needs to license it under the GPL, otherwise that person's licence to use the software will terminate. The full text of versions 2 and 3 of the GPL (both versions are in common use) can be found on the website of the Free Software Foundation.[9]

Potential stifling effect of sharealike licensing

22 When considering whether the second limb of the "sharealike required" exception to the Permissive Licensing Principle applies (the 'genuine need' limb), agencies should take into account the potential of a sharealike licence to:

(a) come across as paternalistic, in that the licensing agency is telling users what they can and cannot do with software that was developed with public funds (i.e., "you can use and develop the software but you can't distribute your developments unless you allow others to do what they want with your developments"); and

(b) have the adverse effect of stifling commercial development of the software by licensees.

[9] See http://www.gnu.org/licenses/gpl-3.0.en.html

23 In theory, sharealike obligations can inhibit commercial development because the licence would require the commercial developer to license its adapted software to others under the same terms when the software is distributed or conveyed. The concern some businesses (and others) have is that this would then enable anyone who obtains a copy of the software to copy, distribute, modify and/or sell it. That, in turn, could harm the developer's commercial interests. In *some* instances this concern may be overstated, in that what customers are really purchasing could be access to the developer's support and maintenance and updates to the software, but this issue is one that divides segments of the open source community. It is for each agency to consider the veracity of such concerns in the context of the software in question and the circumstances surrounding its use.

Security code review

24 Before releasing software for re-use, agencies should:

(a) consider whether there is any prospect that the code or associated files may contain sensitive elements that may need to be removed prior to release; and

(b) if there is such prospect, have the code and any associated files security-reviewed before release.

No additional controls or discrimination

25 When releasing developed software under an open source software licence, agencies must not seek to impose controls or requirements, whether contractual or otherwise, that are inconsistent with the freedoms or permissions granted by the selected OSS licence. For example, agencies are *not* able to license software under an open source software licence and then seek to discriminate between individual, not-for-profit and commercial uses of the software.

No charging

26 Government agencies that release software under open source software licences should not seek to charge people for access to the software.

Updating open source licensed software

27 If an agency:

(a) publicly releases software on open source terms; and

(b) subsequently identifies a bug or other issue with the software that could have a material adverse effect on users of the software,

the agency should (subject to paragraph 28):

(c) consider whether to inform users of the software of the bug or other issue (e.g., by adding a notice to the repository, site or service that contains the software files); and

(d) if the agency has rectified the bug or other issue for its own purposes, release the updated file(s) to the relevant repository, site or service.

28 If the agency is aware:

(a) of a bug or other issue that creates a risk to the security of sites or services using the software or to the confidentiality or privacy of information held by those sites or services; and

(b) that other government agencies are using the software,

the agency should inform the Government Chief Information Officer (and where relevant the Government Chief Privacy Officer and Office of the Privacy Commissioner) as soon as possible, take all reasonable steps to inform the other agencies of the risk and give them time to mitigate the risk before making any public announcement that could result in hackers or others exploiting the bug or other issue.

Code forking[10]

29 Code forking occurs when agencies make changes to the code of open source software without publishing the code back to the software's development community. The fork is the split between the agency's version of the software and the version published by the community. Any further changes made by either the agency or the community will increase the fork. This can make it difficult for the agency to upgrade to a new published version, as the agency would have to reapply all its changes. This risk may be mitigated by contributing modified source code back to the open source software community.

30 Where an agency has taken and modified open source software, it should contribute the modified software back to the open source community unless there is a compelling reason not to do so.

Obtaining rights when procuring or commissioning the development of software

31 When procuring or commissioning the development of software, government agencies should consider whether the software should, in accordance with these Policy Principles, be released to the public for re-use under an open source software licence. If the software should be released to the public for re-use under an open source software licence, government agencies should consider the steps that may be required as part of their procurement and contracting processes to ensure that either:

(a) they have the relevant rights to release the software under an open source software licence; or

(b) that the developer will release the software under a specified open source software licence.

Such steps may include:

(c) ensuring the agency owns the intellectual property rights in the developed software; or

(d) ensuring the agency obtains a broad licence from the service provider allowing the agency to sub-license the software under the MIT licence or, where applicable, the GPL (or, if required, some other open source software licence); or

(e) insisting on contractual provisions that require the service provider to release the software under a specified open source software licence (and, where relevant, to a specified code respository).

Taking these steps may require:

(f) the inclusion of appropriate paragraphs in a notice of procurement (where applicable); and/or

(g) the inclusion of specific contractual provisions in a draft contract; or

(h) where an existing panel supply arrangement is being used, a review of the intellectual property provisions in the panel contract and, if required and possible, amendments to them for the purposes of the software development in question; and

(i) ensuring that the contract does not include confidentiality provisions that would inadvertently prevent release of the software under an open source software licence.

32 Paragraph 31 is subject to any statutory, policy or commercial imperatives to the contrary.

33 The Guidelines for the Treatment of Intellectual Property Rights in ICT Contracts are *not* a policy imperative to the contrary. Rather, they – like NZGOAL-SE – are government policy for agencies to take into account at the time of procuring software. As explained at paragraph 11 above, those Guidelines and NZGOAL-SE are complimentary.

---

[10] This principle is based in part on a discussion of code forking in the Australian Government's *A Guide to Open Source Software for Australian Government Agencies*, above n 1, but has been modified for the purposes of NZGOAL-SE. Most of the Australian Guide has been released under a Creative Commons Attribution 3.0 Australia licence: http://creativecommons.org/licenses/by/3.0/au/.

Act fairly towards developers when drafting IP warranties and indemnities

34 If:

(a) a government agency is commissioning a service provider to develop software that is to be released on open source terms (either by the agency or by the service provider under a contractual obligation for the service provider to do so); and

(b) it is known that the developer will be leveraging or adapting existing open source software developed by others,

the agency should act fairly towards the service provider in relation to the drafting of intellectual property warranties and indemnities.

35 In particular, it is generally considered unreasonable to expect a service provider to give an unqualified IP warranty and an unqualified IP indemnity in relation to third party open source software that the agency agrees can be used by the service provider in developing software for the agency. This is especially so when the agency will be releasing the developed software under an open source software licence or requires the service provider to do so on its behalf.

Liability

36 The MIT licence and the GPL both contain broad disclaimers of warranties and exclusions of liability that are widely known and acknowledged and ought to protect releasing agencies from liability in connection with the software they have released. In essence, people use open source software at their own risk. Agencies should ensure that all disclaimers and exclusions contained in the MIT licence or the GPL (as applicable) are replicated when they release software under these licences.

Review and Release Process

37 Government agencies should follow the NZGOAL-SE Review and Release Process before publicly releasing and licensing their software for re-use under an open source software licence. The Review and Release Process is set out below.

NZGOAL-SE Review and Release Process

Introduction

38 It is recommended that government agencies follow the review and release process set out below before releasing software for re-use under an open source software licence, with assistance where required from their technical and legal teams. The process consists of five main stages:

(a) copyright-related rights evaluation;

(b) evaluation of exceptions;

(c) alternative OSS licensing;

(d) application of the chosen licence; and

(e) release of the software.

39 Each stage contains one or more issues that may need to be worked through. The stages and the issues within them reflect a mixture of the NZGOAL-SE Policy Principles, legal requirements and practical considerations.

40 It can be important to work through these steps to ensure that the agency:

(a) has all relevant rights in or to the software it proposes to release;

(b) uses the appropriate open source software licence; and

(c) does not expose either itself or those who may re-use the released software to liability or related risk.

41 A decision tree diagram for the review and release process is set out at paragraph 62 below.

Stage 1: Copyright-related rights evaluation

What the stage involves

42 The first stage involves:

(a) clearly identifying the boundaries of the software that the agency proposes to release (e.g., the software may comprise a range of files, all of which would need to be released together);

(b) determining whether that software constitutes or contains one or more copyright works; and, if so

(c) evaluating whether the agency has sufficient rights to license the software under the MIT licence in accordance with the Permissive Licensing Principle.

43 In the vast majority of cases, software that an agency wishes to license will constitute or contain one or more copyright works. In the unlikely event that a given piece of software does not constitute or contain one or more copyright works, an agency could, if no exception in Stage 2 applies, release it into the public domain using an NZGOAL-style “no known rights” statement. See NZGOAL for that statement. It is not discussed further in NZGOAL-SE.

Issues where agency does not own all copyright

44 When an agency is in the situation of not owning all copyright in the software it would like to release and license for re-use and needs permission from the copyright owner(s), it is important to appreciate that the software may:

(a) comprise all new code (i.e., be a completely new copyright work);[11] or

(b) build on pre-existing code (i.e., be an adaptation / derivative of pre-existing code).

[11] For example, the software could be a deliverable under a services contract the agency has with a vendor, but the contract may confer copyright ownership on the service provider and only license the software to the agency.

45 The analysis for these two scenarios is different:

(a) **All new code**: In the case of all new code that the agency doesn't own but wishes to license, the agency would need to either:

(i) already be entitled, under an open source software licence that the owner has recently applied to the software, to sub-license the software; or

(ii) obtain irrevocable written permission from the copyright owner to sub-license the software on the relevant open source software licence terms.

Notes on sub-licensing: As to the first option, it is important to note three points. First, if the software has already been licensed under an open source software licence, there may be no need for the agency to sub-license it because anyone who gets the software has the rights granted by the licence that the owner has applied. Second, some open source software licences prohibit sub-licensing. For example, you cannot sub-license GPL-licensed software. Third, if there is a right to sub-license, that right would need to be broad enough to allow sub-licensing under the particular open source software licence that the agency wishes to apply to the software.

(b) **Adaptation / derivative of pre-existing code**: In the case of an adaptation / derivative of pre-existing code, the agency would need to:

(i) be able to identify who owns the pre-existing code (noting that there could be more than 1 owner/contributor) and who has made and owns the new copyright in the adaptation / derivative work; and

(ii) to the extent that the agency does not own the copyright in the adaptation / derivative work, have permission from the other copyright owner(s) for their parts of the overall new work to be licensed under the open source software licence the agency wishes to use.

To understand this, one needs to appreciate that an adaptation / derivative work consists of property (copyright) owned by the original licensor(s) (let's call them A) plus new and separate property (copyright) over the new original parts of the adaptation / derivative work that are created by B. The derivative work is a distinct copyright work in its own right but B doesn't obtain property rights that are greater than B's own contribution. As a United States court has put it, "[t]he aspects of a derivative work added by the derivative author are that author's property, but the element drawn from the pre-existing work remains on grant from the owner of the pre-existing work".[12]

Depending on the number of pre-existing owners, the licences (if any) under which they may have released their code and the licence the agency wishes to apply to the adaptation / derivative work, this can get complex very quickly. In cases of any complexity, agencies may need to seek expert legal advice.

Common scenarios where agency will not be able to license software under the MIT licence

46 There are certain scenarios in which, from a copyright perspective, an agency will not be able to license software under the MIT licence. Three common scenarios are as follows:

(a) **third party owner**: copyright in the software is owned by a third party and the agency is not permitted to license the software for re-use under the MIT licence;

(b) **adaptations of proprietary software**: the software is an adaptation or derivative of proprietary software and the agency does not have the proprietary software owner's written and irrevocable authorisation to license the adaptation or derivative under the MIT licence (whilst the agency might own the copyright in its new contributions, it still needs authorisation from the owner of the base software that it adapted); and

[12] *Stewart v Abend* 495 U.S. 207, 223 (1990).

(c) **adaptations of GPL or similarly open source licensed software**: the software is an adaptation or derivative of open source software that is licensed under a sharealike OSS licence, such as the GPL, that requires distributed or conveyed adaptations or derivative works to be licensed under the same licence.

Stage 2: Evaluation of exceptions

47 If an agency has completed Stage 1 and concluded that it does have sufficient rights to license the software under the MIT licence, then the NZGOAL-SE Policy Principles recommend that the software be licensed with the MIT licence unless an exception set out in paragraph 18 applies.

48 For each proposed release, the exceptions need to be considered in the light of all the surrounding circumstances relevant to the specific software and its release.

49 In many instances, the exercise will be quick as none of the restrictions will apply. In that event, the agency can move to Stage 4 (Application of chosen licence) below. This is because the Permissive Licensing Principle will not have been displaced.

50 If one or more of the exceptions applies, the Permissive Licensing Principle will be displaced and not apply. One of two consequences will follow:

(a) **No open sourcing licensing**: If the exception(s) that apply are any of those in paragraphs 18(a)-18(c) (civil wrongs, commercial interests and/or security or privacy risk), open source licensing of the software will in all likelihood not be possible, unless the software can be amended or altered to remove the reason(s) for the exception(s) applying. If open source licensing is not possible, the analysis ends and there is no need to consider the subsequent stages below. If the software can be amended or altered to remove the reason(s) for the exception(s) applying, the Permissive Licensing Principle is resurrected and one progresses to Stage 4.

(b) **Alternative open sourcing licensing**: if the exception in paragraph 18(d) applies (sharealike required), the software can instead be licensed under an open source software licence with a sharealike requirement. This then raises the question of which of the sharealike OSS licences should be used and that, in turn, takes one to Stage 3 (Alternative OSS licensing).

Stage 3: Alternative OSS licensing

51 Stage 3 only applies where the exception in paragraph 18(d) applies (sharealike required) and no other exceptions apply. If no exception applies, the agency should move to Stage 4 (Application of chosen licence).

52 As noted in paragraph 18(d), a sharealike open source software licence may need to be applied either because:

(a) the agency is required to license the software under a particular sharealike licence (the agency may, for example, be adapting pre-existing software that has been licensed under a sharealike licence like the GPL); or

(b) because the agency has a genuine need for developments to the software to be shared with the developer community.

53 If the agency is required to license the software under a particular sharealike licence, it must license the software under that licence (failure to do so would likely result in the agency breaching third party rights). Where this is the case, the agency should proceed to Stage 4 (Application of chosen licence).

54 If a sharealike open source software licence is to be applied because the agency has a genuine need for developments to the software to be shared with the developer community, the agency is encouraged to use either:

(a) the GPL (version 2 or 3, with version 3 being preferable); or

(b) a sharealike OSS licence that is GPL-compatible.

55 GPL-compatibility is important given the large volume of software that has been released under the GPL. The Free Software Foundation maintains a list of licences it considers to be GPL-compatible. Note that only some of these are sharealike (copyleft) licences.

Stage 4: Application of chosen licence[13]

Introduction

56 Stage 4 explains how agencies apply their chosen open source software licence to the software they're releasing.

Applying the MIT licence

57 This is the standard MIT licence:

> **MIT Licence**
>
> Copyright (c) <year> <copyright holders>
>
> Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
>
> The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
>
> THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

58 Agencies can apply the MIT licence in one of two ways:

(a) They can reproduce the full text of the licence above at the top of each software file they are licensing. When doing so, they need to add the year the software was completed and who the copyright holders are. If the licensing agency is a department of the Crown, it should also replace "Copyright" with "Crown copyright". For example, if the licensing agency were a department, its licensing statement would look something like this:

> **MIT Licence**
>
> Crown copyright (c) 2015, Land Information New Zealand on behalf of the New Zealand Government.
>
> Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell

[13] For a useful general discussion of this topic, see the Software Freedom Law Center's "Managing copyright information within a free software project" at http://softwarefreedom.org/resources/2012/ManagingCopyrightInformation.html

```
copies of the Software, and to permit persons to whom the
Software is furnished to do so, subject to the following
conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES
OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT.  IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT
HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,
WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR
OTHER DEALINGS IN THE SOFTWARE.
```

If the licensing agency is not part of the Crown (e.g., it might be a Crown entity), it would use the term "Copyright" rather than "Crown copyright" and it would state its name without reference to the "New Zealand Government". For example:

```
Copyright (c) 2015, Commerce Commission (New Zealand).
```

(b) Alternatively, agencies can provide the licence as a separate text file (which would contain the full MIT licence text) that accompanies the software files, usually in a top-level folder. If agencies take this approach:

(i) the text file containing all the licensing text should be called "LICENSING" (or something similar); and

(ii) the first paragraph of the licensing text that identifies the licensor should also identify the software to which the licence is being applied, e.g.:

```
SpatialZone Project, Crown copyright (c) 2015, Land Information
New Zealand on behalf of the New Zealand Government.
```

Note that, under this approach, if a single software file is separated from the distribution, the recipient is unlikely to see the applicable copyright notice. If this is of concern to an agency, the solution is to include a brief copyright notice in each file's header that points to the top-level LICENSING file. For example:

```
SpatialZone Project, Crown copyright (c) 2015, Land Information
New Zealand on behalf of the New Zealand Government. This file is
released under the MIT licence. See the LICENSING file found in
the top-level directory of this distribution for more
information.
```

Applying the GPL

59 Full instructions on how to apply can be found on the Free Software Foundation's website.[14]

Other licences

60 If for some reason you are applying an alternative open source software licence, you may wish to check whether the entity that maintains the licence has instructions on how to apply it. Alternatively, you could follow the approach suggested above for the MIT licence or the Free Software Foundation's suggested approach for the GPL.

[14] See "How to use GNU licenses for your own software" at http://www.gnu.org/licenses/gpl-howto.html

Stage 5: Release the software

61 Once the chosen licence has been applied to the software files, release the software for re-use into one or more relevant code repositories and/or on the agency's website. Consider:

(a) using a version control repository like GitHub; and

(b) whether to:

(i) issue a press release about your release of the software; and/or

(ii) notify appropriate communities of interest.

Review and release process decision tree

62 The decision tree diagram below illustrates the Review and Release Process explained above. It is intended to be read in conjunction with the explanations above for each stage.