Master's thesis

Master's Programme in Computer Science

Public licenses in Software Engineering: A Multivocal Literature Review

Akira Taguchi

April 28, 2025

Faculty of Science

University of Helsinki

Contact information

P. O. Box 68 (Pietari Kalmin katu 5)
00014 University of Helsinki,Finland

Email address: info@cs.helsinki.fi
URL: http://www.cs.helsinki.fi/

HELSINGIN YLIOPISTO – HELSINGFORS UNIVERSITET – UNIVERSITY OF HELSINKI

Tiedekunta — Fakultet — Faculty	Koulutusohjelma — Utbildningsprogram — Study programme
Faculty of Science	Master's Programme in Computer Science

Tekijä — Författare — Author

Akira Taguchi

Työn nimi — Arbetets titel — Title

Public licenses in Software Engineering: A Multivocal Literature Review

Ohjaajat — Handledare — Supervisors

Prof. Tomi Männistö

Työn laji — Arbetets art — Level	Aika — Datum — Month and year	Sivumäärä — Sidoantal — Number of pages
Master's thesis	April 28, 2025	26 pages, 25 appendix pages

Tiivistelmä — Referat — Abstract

Context: Public licenses are central to the distribution of works in software engineering. For example in open source there must be an appropriate PCL attached to the source code in order for open-source software to be freely available for possible modification and redistribution. Understanding PCLs can be difficult. This could stem from the legal nature of the license texts and the large number of already-existing PCLs. As a result some actions made within the boundaries of the PCLs may come as a surprise to the public.

Objective: The primary goal of this research is to conduct a multivocal literature review of the current state of PCLs in software engineering, the evaluation of the them and the evidence level of the research. The research aims to provide a novel perspective on relevant licenses and to extract key findings through a rigorous literature review process. This study has two main viewpoints: to provide rigorous research on PCLs to the academic field and to provide insights to the professional field of software engineering on PCLs. The grand goal of this thesis is to raise awareness of the importance of PCLs so that more licensers would make the correct choices based on their situations and needs in a mindful way.

Method: The search strategy examined 6666 sources, found through websites that list PCLs and ad-hoc searches. Applying inclusion and exclusion criteria resulted in the selection of 666 sources, which made relevant contributions related to PCLs in software engineering.

Results:

Conclusions:

ACM Computing Classification System (CCS)
Social and professional topics → Computing / technology policy → Intellectual property → Licensing

Avainsanat — Nyckelord — Keywords

open source, free / libre software, copyright, proprietary software, copyleft, license

Säilytyspaikka — Förvaringsställe — Where deposited

Helsinki University Library

Muita tietoja — övriga uppgifter — Additional information

Software study track

Acknowledgements

much love to artemis, sami nurmivaara, prof männistö and prof mäntylä

thanks to def for borrowing gpt4. thanks to rashid and barunes for sending me software licensing related videos and news. thanks to iikka and joonas for giving a hint for using a license database. thanks to suvi for supporting my work with the thesis from the beginning and enduring my mental health issues.

Contents

1 Introduction

use the word public license instead of PCL, make fixes per perens' other comments. Free software license != strong copyleft software license

PCLs play a central role to the distribution of works in software engineering. For example in open source there must be an appropriate PCL attached to the source code in order for the piece of software to be freely available for possible modification and redistribution. Because open source is central to software engineering the licenses enabling open source must also be considered important in the same context.

Public license is defined by Wikipedia with the following words (Wikipedians, 2024a):

> "A public license is a copyright license where the licensees are not limited. Examples include free content, open content, Creative Commons, free software and open source licences."

Understanding PCLs can be difficult. This could stem from the legal nature of the license texts and the large number of already-existing PCLs. The license texts usually favors correctedness over the readability for the developer. This is because the license text has to act as a valid legal instrument otherwise it cannot be endorsed (Ferguson, 2006). The lack of understanding of PCLs leaves too much room for interpretation. In June 21, 2023 International Business Machines' (IBM) Red Hat seemingly violated a PCL, the GNU General Public License version 2 (GPL-2.0) (Kuhn, 2023) (McGrath, 2023). This was an unpleasant surprise to the public since the project behind GNU General Public License (GPL), GNU Project initially attempted to ensure the users via the GPL have to the following three freedoms (GNU, 1996):

- Freedom 1: The freedom to study how the program works, and change it so it does your computing as you wish. Access to the source code is a precondition for this.
- Freedom2: The freedom to redistribute copies so you can help others
- Freedom 3: The freedom to distribute copies of your modified versions to others. By doing this you can give the whole community a chance to benefit from your changes. Access to the source code is a precondition for this.

Regardless, IBM's Red Hat essentially rendered the previously public Red Hat Enterprise Linux (RHEL) into proprietary software. If the licenses would be more easily understood the proprietarization of RHEL would have been less of a surprise to the users.

On top of PCL details, software engineers in general have a tough time understanding the basic goals of PCLs used in software engineering. In the instance of the RHEL incident it would not have been a big surprise to software engineers if they would have known about other licenses and what they try to achieve or how old is GPLv2 and why it has been succeeded by GNU General Public License version 3 (GPL-3.0).

This thesis' goal is to contribute into the solving these problems in a structured manner. First we state definitions and terminology used in the scope of this thesis. We go over the reasons why there does not exist consistent terminology in this area and why the conversely the definitions are the most stabile ones in this area. Second we take a deep dive into the PCLs through a multivocal literature review. To make more information available, a mapping study connected to the terminology scope defined in the first step is needed. Third includes our own suggestions and basic knowledge for professionals and academics in the industry to enhance the understanding of PCLs in software engineering. This step also includes discussion of the future research and contributes to stablizing the terminology and reinforcing the already-existing definitions in the academic field.

1.1 Research goal, questions and contributions

The primary goal of this research is to conduct a multivocal literature review of the current state of PCLs in software engineering, the evaluation of the them and the evidence level of the research. The research aims to provide a novel perspective on relevant licenses and to extract key findings through a rigorous literature review process. The research questions of the review are:

- RQ1: How many PCLs in software engineering does there exist?
- RQ3: What is the average length of a PCL in software engineering?
- RQ3: What are the most common components seen in PCLs in software engineering?
- RQ4: What are the most common changes made to PCLS in software engineering?

Terms such as open source, source code, software freedom and other vocabulary must be

defined in the scope of this thesis. Section 1.3 will examine this plethora of of terminology and definitions and will be used to establish a sound basis for discussing this broad subject.

This study has two main viewpoints. The first one is to provide rigorous multivocal research on PCLs to the academic field. Because this thesis already does the multivocal work on PCLs in software engineering the researches of the future can cite the results of this thesis without having to mark their study a multivocal one. This is the grand goal of this thesis. The second one is to provide insights and general metrics to the professional field of software engineering on PCLs. Hopefully this makes conversation on PCLs in software engineering easier and more rooted to scientific research rather than gut feeling and old, non-scientific articles on the insights and metrics of PCLs in software engineering.

1.2 Thesis structure

This thesis follows the IMRaD structure. Chapter 1 introduces the problem, this thesis' possible contributions and some further background. Chapter 2 goes over the process and the methods of the multivocal literature review. This is where most of the actual research takes place in. Chapter 3 presents results to the research questions. Chapter 4 discusses implications for research. The chapter also discusses software engineering professionals in the thesis' context and the validity of the thesis' research. Chapter 5 concludes this thesis with the help of the research questions and the future of the research.

1.3 Background and terminology of PCLs

The current terminology is used with different definitions which leads to inconsistencies in the field of software engineering. For example The Open Source Initiative (OSI) classifies GPL-3.0 under the term "open source" whereas the Free Software Foundation (FSF) classifies GPL-3.0 under the term "free software" (OSI, 2008)(Stallman, 2009). This is because their definitions on open source and free software differ from each other. Some parts of the two definitions are even mutually exclusive. This is rarely mentioned when people talk about Free and Open Source Software (FOSS) or Free / Libre and Open Source Software (FLOSS) which leads to misunderstanding that the two approaches are the same. This is why our focus will be PCLs in software engineering, which distinguishes our investigation from the broader topic of PCLs or the copyright law. This includs also PCLs that are not approved by the FSF nor OSI hence not falling under the group of FLOSS licenses. The

term ”copyleft” is defined by Mustonen, 2003 in the following way:

> ”Copyeft is a novel licensing scheme. It facilitates open and decentralized software development. Its key feature is that once a program is licensed by the inventor, the subsequent programs based on the original must also be licensed similarly.”

This is why the term is often used in the context of free software.

In this section we aim to increase the accessibility of our discussion by providing a concise overview of the background of the field of PCLs and the terms we employ.

To explain our emphasis on PCLs in software engineering, it is essential to examine the other possible areas of interest in PCLs. Our study classifies such efforts into eight domains as mentioned by the GNU Project (GNU, 2023).

These domains include:

- PCLs in software engineering
- PCLs in documentation for example architecture documentation of a project that may or may not be software or even publicly licensed
- PCLs in artistic works for example digital art, music or videos
- PCLs in educational works
- PCLs in fonts
- PCLs in viewpoints
- PCLs in physical objects
- PCLs in other works

The primary aim of this study is to investigate PCLs in software engineering process. However, it is important to acknowledge that PCLs in software engineering are only aspect of PCLs. These additional dimensions are crucial in adoption and implementation of PCLs in software engineering, but they are not the focus of this thesis.

For example, including artistic works such as music would require us to understand the basics of music theory and what sets apart distinct pieces of music from one another,

something that could be outside the skillset of the author. While developing a comprehensive theory, framework, and tooling for PCLs as a whole is a gargantuan task beyond the scope of a single thesis, narrowing our focus to software engineering enables us to examine a more concise and complete aspect of the main topic of this thesis.

As significant point of clarification, it is essential to acknowledge that PCLs are generally meant to be used as valid legal instruments. The question whether or not a PCL can act as a legal instrument is critical to the main function of these licenses. However, this thesis will not focus on the legal doctrine aspects either. The enforceability of PCLs has seen discussion in the academic field of law since the dawn of PCLs and since there's already an academic base for research it is likely the discussion seems to continue on with a healthy amount of activity (Duisburg, 2011).

Since the most recognized PCLs in software engineering in public are either open-source licenses or free-software licenses and since both paradigms are driven by different organizations with very different goals and values, it is understandable how non-standardized the terminology in the scope of PCL in SE is. The example given in the first section of this sub-chapter illustrates the challenges involved in maintaining consistency in the use of terminology in this emerging field and further warrants a closer inspection of the terminology to emphasize our own standing in the field.

To provide an understanding of the terminology used in this thesis, a Venn diagram is presented in Figure 1.1, which contextualizes the non-standardized terminology within the PCL scope as a whole. This perspective provides an increased understanding of where different subdomains fall in the larger picture of PCLs. Furthermore it is essential to note that PCLs in software engineering encompasses different aspects that require a closer examination.

Let us explore further the differences and similarities between open source and free software at the software engineering level of PCLs. This is a crucial step since we can see from the approximation in Figure 1.1 that the majority of PLCs are either free software, open source or both. We glanced over the free software definition in the first section of Chapter 1. Open Source Initiative defines open-source licenses in the Open Source Definition briefly in the following way (OSI, 2024):

> "Open source licenses are licenses that comply with the Open Source Definition - in brief, they allow software to be freely used, modified, and shared."

Like the FSF with free software, OSI has the final word on what passes as open source

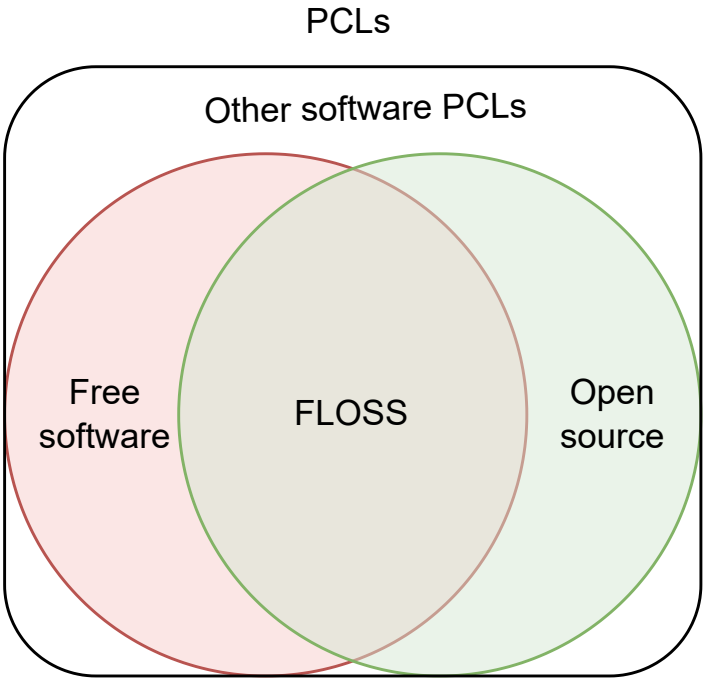

Figure 1.1: PCLs in software engineering

and what does not. For example a new software PCL will not classify as free software nor open source until the corresponding organization has acknowledged the software PCL as either free software, open source or neither. If a PCL is accepted by both FSF and OSI it will fall under the term FLOSS. If a PCL gets accepted by neither of the organization or it gets rejected by both organizations it will fall under other software PCLs in the Figure 1.1. In general free software license requirements are considered more strict than the open source license requirements. For the sake of perspective we could simplify the differences like so: free software requires the redistributions of the licensed software to be open as well but open source does not require this. The terms free software and open source are in general often misunderstood or just thought of as FLOSS collectively because the terms have a hard time conveying their paradigms in the natural language. One would not think free software does not mean software free of charge nor would one think that open source allows closed source redistributions of the licensed software. We will glance over the impacts on the industry of these two terms in Chapter 4.

With the context laid out in this chapter let us define PCLs in software engineering for the purpose of this study: Public software licenses are copyright licenses where the licensees are not limited and the copyright license in question is meant be used in licensing software source code. This helps us create the search strings and find the relevant literature for

this thesis. This also helps us exclude PCLs regarding documentation, media and all other non-software targeted PCLs.

The quest to categorize every software PCL under some paradigm objectively is a complex one and cannot be comprehensively answered in a single paragraph. Therefore it is essential to continue taking the correct steps towards incresing the scientific understanding and providing the industry with examples, standards and processes to follow. However, as the following chapters reveal, a significant amount of effort is still being spent on solving the same problem multiple times, rather than building on existing knowledge and finding the next problem to solve. This thesis aims to contribute to mitigating this challenge by providing a rigorous analysis of the current state of the field. As the knowledge, conventions, and terminology take shape,we can look forward to reaching a state where less effort is spent on defining concepts and more on practical problem-solving.

2 Methods

This chapter aims to establish a precisely defined and rigorous research approach to enhance transparency and repeatability. We will take the steps required to ensure that every phase and decision is thoroughly documented, enabling the reader to retrace the research process. In a thesis made by a single researcher the lack of cross-examination of results with multiple researchers and the validation of evaluation criteria for opinion bias pose threats to validity, as will be clarified further in Chapter 4. Therefore, special attention will be paid to address these concerns. By following this approach, this research endeavors to contribute to the existing body of knowledge in the field of computer science in a robust and reliable manner.

at the end of this section i will have to document the timestamp at which point i accessed the docs of scancode-licesendb since. actually i might be a good idea to create a wayback machine attempt to capturing the state of the licensedb rn. nvm great wayback machine is not viable solution for that. just stating here that i downloaded the licensedb on 2025 mar 25 15:30

it's good to note that when bumping into the missing license of attpubliclicense which was from gnu, it turned out that the license listing site doesnt state the license content whatsoever. i had to just put the comment of the license into manual licenses.

manual/missing licenses: check which shortcodes of missing licenses does stage2-3.py output. check from sheets where does the shrotcode originate from. ctrl + f that listing site. copy the full license to manual-licenses/shortcode.txt and re-run the stage2-3.py until no longer missing licenses appear in console.

remember to count here how many missing licenses / manual licenses had to be done manually

could be also a good idea to note here how many missing licenses came from which site. gnu and fsf seem to be the winners here.

cvw was not found with the python script. i checked from the excel that its from the osi but osi didnt have it anymore by the time i got there. at some point between fetching the data and the re-fetching the license content from the original site MITRE had voluntarily retired the cvw license. it was found from the scancode licensedb with the name cvwl instead.

dejavu and dbg-3.0 were also two other licenses that contained a space. this might indicate that the space is an accident that its simply just not found from a license listing site x. its also good to note that the python script was decided to be an valid approach since many of the licenses were actually found with the shortcode from the licensedb scancode. fetching 700 licenses by hand would have had time and validity issues. wayback machine could have been used to do the actual searching as well. this is unfortunately a validity issue but at least the source is available in wayback machine.

let it be said, that when looking for manual licenses, if there existed two sites which listed the same license, the leftmost was chosen, as per the order of the wikipedia mit article

it seems like i have manually added licenses and invented shortcodes for them AT LEAST in DFSG license listing. FAL is a good example.

For example licenses like CorkForkPL from FSF are just empty licenses. CorkForkPL is used however in MighTyD project but the license would have to be seen from a downloaded project or something like that. A new scope: only licenses that are one (1) click away from the initial license landing page can be copypasted to the manual licenses. just like the JahiaCSL has a URL on FSF for the license new location although the FSF page is empty.

licenses like MPL exist on FSF and GNU. it was not easily found from FSF (empty with links to programs using this) so it was gotten from GNU which was labeled as MPL1.1, which the FSF DOES have so i just boldly went with that. threat to validity.

it could be a good idea to mention how many missing licenses came out of which sites. or it could be out of scope. i can just make a validity threat and say with face value that most of them were from FSF, GNU in that order. Python license seems just straight up an accident on FSF's side. scope is not to fix the documentation problems of the 5 organizations though so I'll leave it just like that and mention the possibility of it being just an accident.

so. now i figured ill take another validity threat L and decide to remove duplicates by human eyesight and choose the one left over from two seeming duplicates by pure human eyesight. i will document the ones that were chosen over the other duplicate and call it a day. for now. i gotta write number + shortcode to find duplicates and document them.

it would be good to note in conclusions that while im stripping away here the errorish licenses and "duplicates". the ones used mostly are different from those that are actually legally valid historically or even those that are not meant to be legally valid (MIT). whilst

im doing this systematically it doesnt mean that all parts of it can be done automatically or by automation. human touch in for example checking the incl excl criteria and removing duplicates by combining and creating the weight value of shortcode + license name + license text is done basically by a human with the help of python first sorting them based on their word overlap (not sort, not full on difflib, not cosine similarity). ill do exclusion first though. inclusion is kinda implied here i just realized when i took them out of the license listing cites and removed blanks etc.

The systematic literature review method (SLR) is a well-established approach for conducting a comprehensive and rigorous analysis of the existing research on specific research question or subject (Kitchenham and Charters, 2007). This paper presents a multivocal literature review (MLR). MLR is a SLR that includes both academic (AL) and grey literature (GL). This method was selected for this study to facilitate a thorough and scientifically interdisciplinary examination of PCLs in software engineering. The existing literature consists of PCLs and as such are considered gray literature, making the thesis a multivocal literature review.

This study follows the guidelines outlined by Kitchenham and Charters, 2007, to ensure its quality. The multivocal review method consists of three distinct phases: planning, conducting and reporting the review. This study stricly adhered to this structure. The phases can be further broken down into a research protocol, as illustrated in Figure 2.1. Adhering to the protocol is the first step in ensuring a well-documented and rigorous process, which increases the validity and auditability of the study.

The multivocal literature review process began with the formulation of research questions and the establishment of a comprehensive search strategy and scope. The search process was conducted by employing a quasi-gold standard (QGS) approach based on the implementation by Zhang and Ali Babar, 2010. After the completion of the search process, the inclusion and exclusion criteria were defined. To ensure a structured evaluation of the literature, a data extraction form was created. Finally, a strategy for analyzing the extracted data from the literature was designed.

To ensure the reliability and validity of the research protocol, it was validated against similar systematic literature reviews in computer science, the aforementioned guidelines by Kitchenham and Charters, 2007, and was further refined through an iterative process. Specifically, a subset of the data was tested on (The QGS) and any identified issues or problems were recorded and addressed. The details of this process are explained and thoroughly documented in the following sections. Similarly, the same approach was followed

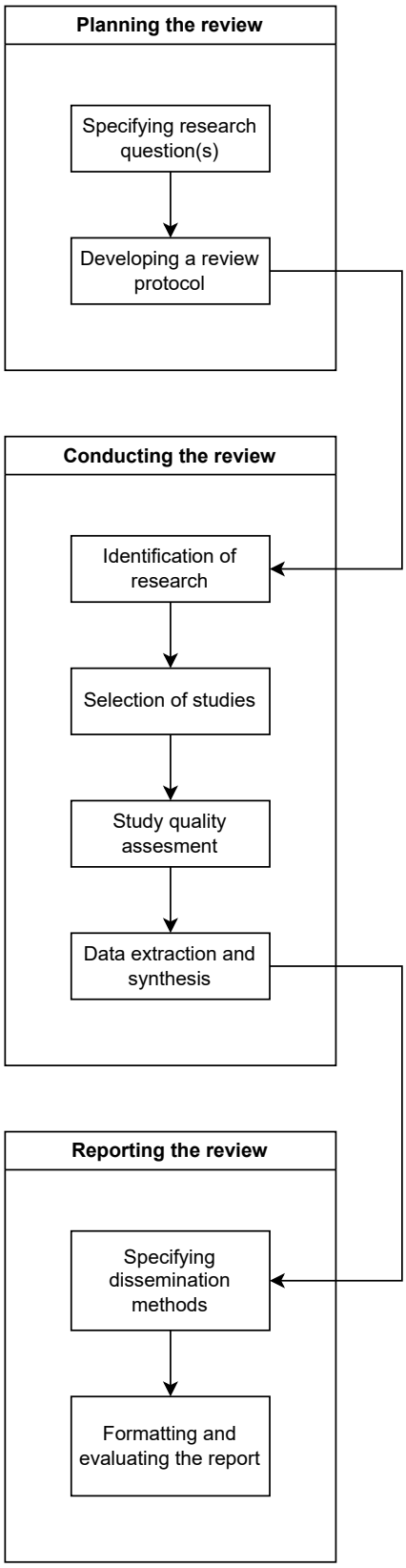

Figure 2.1: Three phases of a systematic literature review

for the data extraction process, whereby a subset of literature was tested to refine the data extraction form. The revision of the form was undertaken as necessary to guarantee the completeness and accuracy of the extracted data.

2.1 Research questions

The research questions in this study served two primary purposes. Firstly, they aimed to provide an anaylsis of the existing multivocal literature on PCLs in software engineering for the researchers interested about the field. Secondly, the questions were designed to cater a secondary audience of professional software engineering practicioners. As discussed in the Chapter 1, the following research questions were addressed in this thesis:

- RQ1: How many PCLs in software engineering does there exist?
- RQ2: What is the average length of a PCL in software engineering?
- RQ3: What are the most common components seen in PCLs in software engineering?
- RQ4: What are the most common changes made to PCLS in software engineering?

The multivocal literature review in this thesis begins with addressing RQ1, which aims to provide the amount of PCLs that exist in software engineering. The review takes into account attributes like versions, supersedences to a different license family, formal or otherwise and recognizability. These attributes give us different amounts to existing PCLs in software engineering. This information could be most valuable for the practicioners out of all the research questions in the thesis since it could give some sense of the scale when picking a PCL that would serve the practicioners' needs the best.

Next RQ2 seeks to find the average length of the text of a PCL in software engineering. This research question has attributes like the number of characters, sentences, distinct sections and the size of the license on a computer screen. This information could be valuable for the practicioners mentioned in the previous parapgrah for the same reasons of getting a better overview of the PCLs in software engineering. The research questions could also be beneficial for the practicioners working directly within the meta plane of PCLs in software engineering. Let us refer to the latter as researchers.

Finally RQ3 and RQ4 attempt to distinguish the top level paragraphs and other components of the PCLS in software engineering and what are the common reasons for the

changes made to them throughout the years. The research questions go over the content of the changes and the implied and expressed reasons for making the changes. The answers to these last two research questions could again be useful for the researchers. The results can be used to introduce some notable background of the current PCLs in software engineering and enabling focus to more specific areas inside this PCLs in software engineering.

2.2 Search stragey

The search process was conducted on various PCL listing websites. The selection criteria for the literature were defined after the search process and the selection process was based on inclusion and exclusion criteria. The inclusion and exclusion criteria and each step of exclusion on the literature found was documented and is available as Appendix A. The used criteria are presented later in this chapter.

The data extraction process was performed in a standardized and systematic manner, with the aim of obtaining all relevant information from the selected literature. The data extraction form used included information such as license name, release year, text length and inferred purpose and is available Table 2.2. The extracted data was then used to answer the research questions and perform the data analysis. The results of the data analysis were then reported in a rigorous manner.

2.2.1 Search method

The search was conducted on various PCL listing websites, as mentioned earlier, to obtain a broad set of multivocal literature. This approach yielded a large number of literature that were processed to a subset of high-relevance literature using exclusion and quality criteria presented later in this chapter. Manual searching of databases with thousands of PCLs is not feasible, and it is prone to researcher bias and may overlook relevant venues from other scientific disciplines. However, a preliminary manual search was performed to reduce the number of iterations required and establish the quasi-gold standard (QGS) mentioned earlier.

Field	Value
Publisher	Massachusetts Insitute of Technology
SPDX identifier	MIT
Debian FSG compatible	Yes
FSF approved	Yes
OSI approved	Yes
GPL compatible	Yes
Copyleft	No
Linking from code with a different license	Yes

Table 2.1: MIT License Wikipedia page infobox

2.2.2 Search scope and terms

Originally the search terms would have been present just like in a normal MLR or SLR. Keywords however produced highly varying and non-reproducabe results in Google Scholar and Google Search. Some PCL listing websites such as FSF's list of pages categorized as licenses could not be found from Google Search even with the `site` operator: `site:https://directory.fsf.org/wiki/Category:License`. Although the page has been up since 2013 for some reason Google has not crawled the page in 10 years (FSF, 2024). Hence why this thesis does not include search terms per se.

Instead, for establishing a QGS we started defining our search scope from the Wikipedia page of one of the most used open source license according to Balter, 2015, the MIT license (Wikipedians, 2024b). The infobox contained fields in the order shown in Table 2.1.

As we defined PCLs in software engineering as copyright licenses where the licensees are not limited and the copyright license in question is meant be used in licensing software source code in Chapter 2 and our research questions focus on finding measurements and reasonings to the PCLs' various attributes, we decided to gather PCLs from the related web pages of the aforementioned categorizers: SPDX, FSF, OSI and GNU. The publisher, GPL compatibility, copyleft and the linking exception did not result in any meaningful PCL listing websites. This leaves us with the SPDX, Debian FSG compatibility, FSF and OSI from which all resulted in some sort of PCL listing websites.

With the search for the initial PCL listing websites completed we moved onto the search process itself.

2.3 Search process

The literature selection process was divided into multiple stages, as outlined in Figure 2.2. The initial step involved the formation of the first PCL listing websites through which the first literature would be acquired from.

does this figure 2.2 have to use these terms since im really just applying K & C 2007 slr guidelines to a mlr. no it doesnt. change it when finishign up 2. methods.

In the first stage, the search was conducted using the ”SPDX License List” (Linux Foundation, 2024), ”The DFSG and Software Licenses” (Debian, 2024), FSF’s ”Category:License” Wiki page (FSF, 2024), GNU’s ”Various Licenses and Comments about Them” (GNU, 2023) and "OSI Approved licenses" (OSI, 2024). The PCLs appear in the same order as decribed above: SPDX, DFSG, FSF, OSI and GNU. The appendix was also crafted in a spreadsheet software so that only the initial hit source was documented in the order described above. For example even if MIT license would be found on SPDX and DFSG Appendix A would only display MIT license with the ”First hit from” value being SPDX. The initial list of 789 PCLs excluding duplicates is provided in Appendix A.

Some things must be mentioned about the process of the first stage. First, the FSF outputted a ”license” named ”other”. This ”license” included at the time of observation 5282 known programs to FSF whose PCLs were not documented yet by the FSF. Although some of the programs had straightforward PCLs such as GPL-2.0-only we decided to leave these PCLs out of the scope of this thesis due to the large amount of the programs. The second note is about GNU’s PCLs. Since we had the most trouble scraping the identifiers automatically from this website we decided to limit the PCLs only to ”Software Licenses” as defined by the table of contents on the website.

approx duplicates were the result of going to the two listing websites that had the approaximately same looking licenses. then i just checked if they were actually some sort of duplicates of one another or if they already exist somewhere else. examples here. this is also a validity threat. problem with focusing software specific licenses is for example wtfpl. it is mostly used in software licensing but it doesn’t quite clearly state that it is software specific license. maybe ill have to include the word "public license" and just include stuff that’s not actually software specific or maybe ill make some exclusion criteria in order to get less non-software licenses

In the second stage, the inclusion and exclusion criteria were applied to further filter the

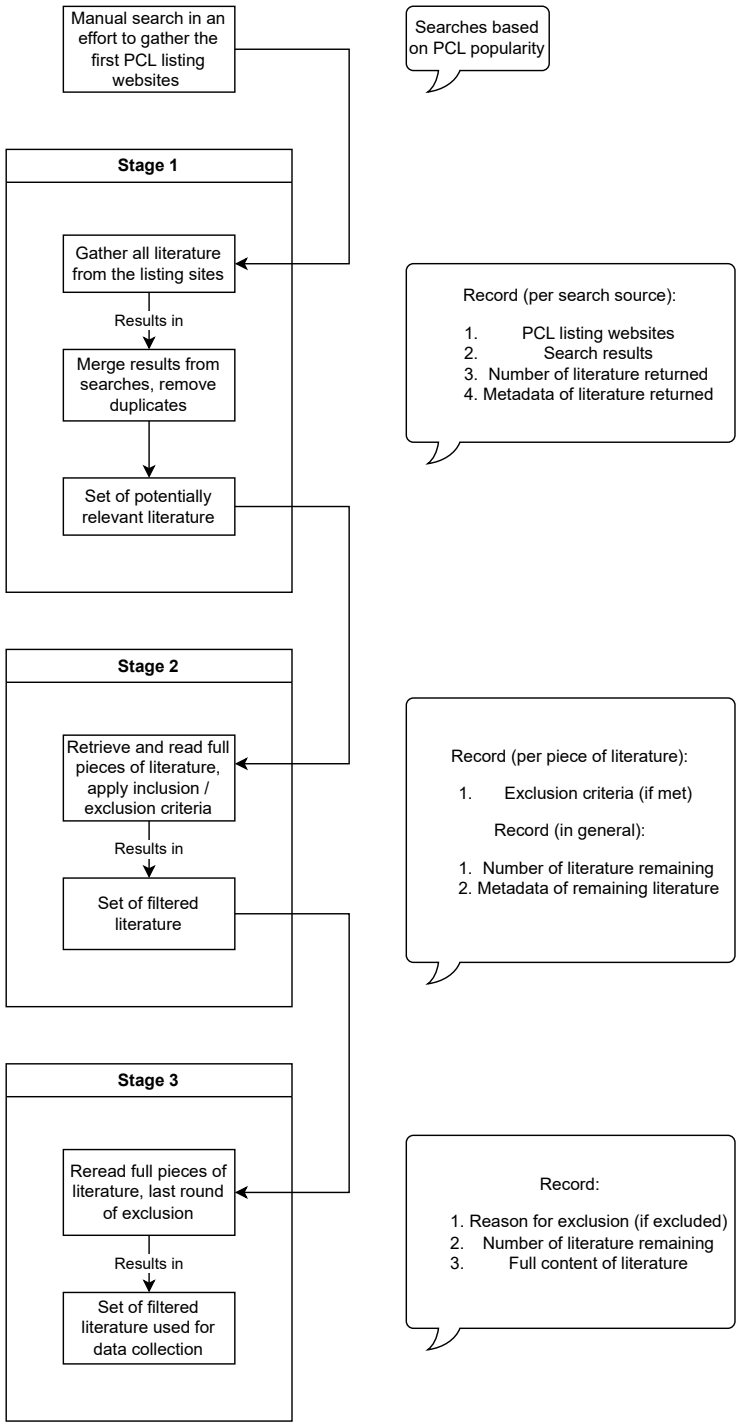

Figure 2.2: Search process divided into stages

literature and reduce the number of licenses to be reviewed. This involved a manual review of the full licenses. The exclusion reason as a shortcode (e.g. I1 = failed to meet inclusion criteria 1 or E2 = met exclusion criteria 2) is provided in Appendix B.

The third stage was the most time-consuming and involved a manual review of the full licenses. After reading and evaluating each license, a final round of exclusions was completed and documented. The remaining licenses were used for data collection and analysis in the final part of the study. The final list of licenses is available in Appendix C.

2.4 Inclusion and exclusion criteria

To be eligible for the data collection and analysis, a license had to meet all of the following inclusion criteria:

- I1: The license focuses on the copyright of software source code or their binaries
- I2: inclusion criteria 2

Additionally, licenses were excluded if they met any of the following criteria:

- E1: The piece of literature is a license exception
- E2: The piece of literature is a Creative Commons license
- E3: exclusion criteria here
- E4: exclusion criteria here

The relevance of each piece of literature was evaluated based on inclusion and exclusion criteria stated above. In cases where there was doubt about the suitability of a license, a more in-depth manual examination of its content was performed. The reason for exclusion was documented for each license that failed to meet the criteria, and when it was unclear, the license was included by default.

Another relevant criteria related to the ones of inclusion and exclusion are the quality and evidence criteria. These criteria used by Dybå et al., 2007 were not put into practice in this thesis since individual PCLs per se might not be meaningful in a results, evidence nor quality perspective. This puts more emphasis on the inclusion and exclusion criteria so that is something we must be mindful about.

#	Field	Concern/Research question
F1	Name	Documentation
F2	Length	RQ3
F3	FSF approval	Documentation
F4	OSI approval	Documentation
F5	Inferred purpose	RQ1, RQ2, RQ4

Table 2.2: Data extraction form

2.5 Data collection and data analysis

To answer the research questions of this thesis, a thorough examination of the selected primary literature was conducted and the necessary data was collected using data extraction form presented in Table 2.2. A record of extracted data was kept for anaysis and is available as Appendix B.

The subsequent chapter presents the outcomes of the steps taken in the study, as discussed above.

3 Results

This chapter employes the data extracted from the set of primary literature, available as Appendix A, utilizing the methods outlined in Chapter 2 to address the research questions. Firstly, a summary of the general statistics collected and aggregated from the studies is presented. Following that, an analysis of the data is performed to provide answers to each of the research questions.

mention design science, the artifact and its contributions

how many licenses and why

statistical overview with figures (mapping study)

how many licenses during each stage (figure)

basic statistic on final licenses (figure)

essential statistics (figure)

3.1 Placeholder question (RQ1)

figures and literature identifier tables

3.2 Placeholder question (RQ2)

figures and literature identifier tables

3.3 Placeholder question (RQ3)

figures and literature identifier tables

3.4 Placeholder question (RQ4)

figures and literature identifier tables

4 Discussion

indications

follow-up observation

observation 1

observation 2

sum-up from those two

4.1 Implications for research

how to improve scientific scene 1

how to improve scientific scene 2

how to improve scientific scene 3

4.2 Implications for software engineering professionals

how to improve professional scene 1

how to improve professional scene 2

how to improve professional scene 3

overall

4.3 Limitations and threats to validity

The major limitation of this study is that the subjective results could not be validated by multiple researchers. In a systematic review, it is standard practice and highly recommended to have at least two, if not more, individuals independently conduct the review processes and then cross validating the findings. This would result in the possibility of

comparing individual exclusion decisions and other decicions, thereby increasing the credibility of the study. However, in this study, the methodology was thoroughly documented, which allows us to assert with confidence that the study has an appropriate level of of validity.

As a work of single researcher, there is also a chance of inaccuracy and bias in the literature selection and filtering process. As much of the literature had to be reviewed manually and then included/excluded on a qualitative basis, this is a known limitation and a threat to validity. Multiple rounds of documented filtering and a clear paper trail of all decisions made keeps this threat in the acceptable levels.

4.3.1 Limitations of literature selection for review

Efforts were made to ensure the inclusion of comprehensive set of literature in the search process. This was achieved by setting the starting point of PCL lists to the Wikipedia article of the MIT license.

However, as with all systematic literature reviews, a comprehensive manual review of all literature would have been a formidable task. Therefore, additional filtering was conducted. This filtering was carried out in two phases, starting with the application of inclusion/exclusion criteria, followed by a second phase focused on evaluating the nature of the PCLs and conducting a manual review. As a result of this second phase, a set of literature were excluded following a critical appraisal, with documentation and reasoning provided for each section.

The first phase of filtering has some notable limitations starting with the two PCL listing websites: SPDX and DFSG. Since the material was gathered to a spreadsheet program the duplicates were removed using the short identifier the listing page was using. Let's look at this validity threat using an example. Suppose our spreadsheet program has acquired the PCL with an identifier "MIT". The results of phase 1 will not include any other PCL marked with the identifier "MIT". In the worst case the identifier "MIT" could have actually been "MIT-DFSG-edition" but with the identifier of "MIT". Since there were so many PCLs in phase 1 it would not have been possible to check the uniqueness of all removed duplicates. One of the reasons why this would not have been feasible is that the listing sites would fetch the PCL contents from another webpage or at the second worst case, from another website. The worst case is that the URl is dead and we get HTTP 404. The amount of PCLs, duplicates and the lack of already existing tools makes this problem

multilayered. However this is the integrity level we decided to live with.

FSF's PCL listing introduced us to pick another limitation for the scope of this thesis. The license shortcoded as "other" was not a PCL but instead a hyperlink to another listing webpage that listed programs that the FSF has no yet managed to document the license which the program uses. Although the one of the programs called "babl" was licensed as with "gplv3" the amount of undocumented programs was over 5200 at the time of observation. For this reason we are excluding the PCLs found indirectly from the category "other".

tell about the validity threats of osi literature selection for review

Lastly, GNU project's listing site allowed us to use a shortcut of sorts which we will document here for the purposes of acknowleding the limitations of it. The table of contents at the listing site marked certain consequtive PCLs as software PCLs. On top of this the PCLs were not organized into easily processable tables but rather in stacked on one another in rich text format. Although we decided to use regex on the HTMl file the included PCLs were only the ones that were simply under the header "Software licenses". In the worst case scenario GNU project could have misinterpreted some PCLs as non-software licenses thus making this thesis exclude them with a wrong reason. While from a quick glance and the existence of the other four PCL listing sites, we think it is still worth documenting when it comes to validity and the integrity of this thesis.

On top of too heavy filters we would also like to document the too light filters in the literature selection for review. We can see from Appendix A that for example PCLs with the literature identifiers L777 and L780 are almost the same regarding the shortcoded identifiers: "ZPL - 2.1" and "ZPL-2.1". The duplicate removal would have been seemingly simple to execute on phase 1. However with the presence of over 700 pieces of literature we decided not to give special treatment to any potential set of duplicates. While it is most possible that OSI's "ZPL - 2.1" is equivalent exactly to SPDX's "ZPL-2.1" we could not be sure without looking at their contents. This could have resulted duplicate PCLs in the literature selection for review but these type of duplicates are removed in phases 2 and 3 due to the PCLs being read in full.

As such we can note that the literature selection was done in a sufficient manner.

4.3.2 Limitations in data extraction

importance of data extraction

lack of measurements and tooling

5 Conclusions

primary objective of this study

conclusions from each rq

5.1 Future research

adopting a clear baseline

why agplv3re is the best license

Docker CLA, SSPL

make cla easier maybe with gpg / joplin easy cla sign

LICENSE highlighting.js

what kind of efforts and why

what this thesis has provided

Bibliography

Balter, B. (2015). *Open source license usage on GitHub.com - The GitHub Blog*. https://web.archive.org/web/20240409120258/https://github.blog/2015-03-09-open-source-license-usage-on-github-com/. Accessed: 2024 April 9.

Debian (2024). *The DFSG and Software Licenses*. https://web.archive.org/web/20240415104532/https://wiki.debian.org/DFSGLicenses. Accessed: 2024 April 15.

Duisburg, H. von (2011). "The Enforceability of the General Public License in the US". In: *Zeitschrift der Deutsch-Amerikanischen Juristen-Vereinigung e.V. 36 (2011)* 2, pp. 69–70. URL: https://heinonline.org/HOL/Page?handle=hein.journals/dajvnws2011&id=75.

Dybå, T., Dingsøyr, T., and Hanssen, G. (Oct. 2007). "Applying Systematic Reviews to Diverse Study Types: An Experience Report". In: pp. 225–234. ISBN: 978-0-7695-2886-1. DOI: 10.1109/ESEM.2007.59.

Ferguson, D. (2006). "Syntar Errors: Why Version 3 of the GNU General Public License Needs Debugging". In: *North Carolina Journal of Law & Technology* 7.2, pp. 397–420. URL: https://heinonline.org/HOL/Page?handle=hein.journals/ncjl7&id=403.

FSF (2024). *Category:License - Free Software Directory*. https://web.archive.org/web/20240409111815/https://directory.fsf.org/wiki/Category:License. Accessed: 2024 April 9.

GNU (1996). *What is Free Software?* https://web.archive.org/web/19980126185518/https://www.gnu.org/philosophy/free-sw.html. Accessed: 2024 Feb 8.

– (2023). *Various Licenses and Comments about Them*. https://web.archive.org/web/20240226115940/https://www.gnu.org/licenses/license-list.html. Accessed: 2024 Feb 26.

Kitchenham, B. and Charters, S. (Jan. 2007). "Guidelines for performing Systematic Literature Reviews in Software Engineering". In: 2.

Kuhn, B. M. (2023). *A Comprehensive Analysis of the GPL Issues With the Red Hat Enterprise Linux RHEL Business Model*. https://web.archive.org/web/20240205080551/https://sfconservancy.org/blog/2023/jun/23/rhel-gpl-analysis/. Accessed: 2024 Feb 5.

Linux Foundation (2024). *SPDX License List*. https://web.archive.org/web/20240412103557/https://spdx.org/licenses/. Accessed: 2024 April 12.

McGrath, M. (2023). *Furthering the evolution of CentOS Stream*. https://web.archive.org/save/https://www.redhat.com/en/blog/furthering-evolution-centos-stream. Accessed: 2024 Feb 5.

Mustonen, M. (2003). "Copyleft—the economics of Linux and other open source software". In: *Information Economics and Policy* 15.1, pp. 99–121. ISSN: 0167-6245. DOI: https://doi.org/10.1016/S0167-6245(02)00090-2. URL: https://www.sciencedirect.com/science/article/pii/S0167624502000902.

OSI (2008). *GNU General Public License version 3*. https://web.archive.org/web/20240226075409/https://opensource.org/license/gpl-3-0. Accessed: 2024 Feb 26.

– (2024). *Licenses - Open Source Initiative*. https://web.archive.org/web/20240307121412/https://opensource.org/licenses. Accessed: 2024 March 7.

Stallman, R. (2009). "Viewpoint Why "open source" misses the point of free software". In: *Commun. ACM* 52.6, pp. 31–33. ISSN: 0001-0782. DOI: 10.1145/1516046.1516058. URL: https://doi.org/10.1145/1516046.1516058.

Wikipedians (2024a). *Category:PCLs — Wikipedia, The Free Encyclopedia*. https://web.archive.org/web/20240131085240/https://en.wikipedia.org/wiki/Category:Public_copyright_licenses. Accessed: 2024 Jan 31.

– (2024b). *MIT License - Wikipedia*. https://web.archive.org/web/20240411081352/https://en.wikipedia.org/wiki/MIT_License. Accessed: 2024 April 11.

Zhang, H. and Ali Babar, M. (2010). "On searching relevant studies in software engineering". In: *Proceedings of the 14th International Conference on Evaluation and Assessment in Software Engineering*. EASE'10. UK: BCS Learning & Development Ltd., pp. 111–120.

Appendix A Primary literature identified in the search process, duplicates removed

Table A.1: A list of literature and the basic filtering step

Literature identifier	Identifier	SPDX	DFSG	FSF	OSI	GNU
L1	0BSD	SPDX			OSI	
L2	996			FSF		
L3	AAL	SPDX			OSI	
L4	Abstyles	SPDX				
L5	AcademicFreeLicense					GNU
L6	ACDL-1.0			FSF		
L7	ACEL			FSF		
L8	AdaCore-doc	SPDX				
L9	Adobe-2006	SPDX				
L10	Adobe-Display-PostScript	SPDX				
L11	Adobe-Glyph	SPDX				
L12	Adobe-Utopia	SPDX				
L13	ADSL	SPDX				
L14	AFL-1.1	SPDX				
L15	AFL-1.2	SPDX				
L16	AFL-2.0	SPDX				
L17	AFL-2.1	SPDX				
L18	AFL-3.0	SPDX		FSF	OSI	
L19	Afmparse	SPDX				
L20	AGPL					GNU
L21	AGPL-1.0-only	SPDX		FSF		
L22	AGPL-1.0-or-later	SPDX		FSF		
L23	AGPL-3.0-only	SPDX	DFSG	FSF	OSI	
L24	AGPL-3.0-or-later	SPDX		FSF		
L25	Aladdin	SPDX		FSF		GNU
L26	Aladdin-9			FSF		
L27	AMDPLPA	SPDX				

L28	AML	SPDX				
L29	AML-glslang	SPDX				
L30	AMPAS	SPDX				
L31	ANTI-1.3			FSF		
L32	ANTI-1.4			FSF		
L33	anticapitalist					GNU
L34	ANTLR-PD	SPDX				
L35	ANTLR-PD-fallback	SPDX				
L36	Apache-1.0	SPDX		FSF		
L37	Apache-1.1	SPDX		FSF	OSI	
L38	Apache-2.0	SPDX	DFSG	FSF	OSI	
L39	apache1					GNU
L40	apache2					GNU
L41	APAFML	SPDX				
L42	APL-1.0	SPDX			OSI	
L43	App-s2p	SPDX				
L44	APSL-1.0	SPDX		FSF		
L45	APSL-1.1	SPDX		FSF		
L46	APSL-1.2	SPDX		FSF		
L47	APSL-2.0	SPDX	DFSG	FSF	OSI	
L48	apsl1					GNU
L49	apsl2					GNU
L50	APSLv1.x			FSF		
L51	Arphic-1999	SPDX				
L52	Arphic-PL			FSF		
L53	Artistic-1.0	SPDX		FSF	OSI	
L54	Artistic-1.0-cl8	SPDX				
L55	Artistic-1.0-Perl	SPDX		FSF	OSI	
L56	Artistic-2.0	SPDX	DFSG	FSF	OSI	
L57	ArtisticLicense					GNU
L58	ArtisticLicense2					GNU
L59	ASWF-Digital-Assets-1.0	SPDX				
L60	ASWF-Digital-Assets-1.1	SPDX				
L61	ATTPublicLicense					GNU
L62	Baekmuk	SPDX				

L63	Bahyph	SPDX				
L64	Barr	SPDX				
L65	bcrypt-Solar-Designer	SPDX				
L66	Beerware	SPDX				
L67	BerkeleyDB					GNU
L68	Bitstream Font License			FSF		
L69	Bitstream-Charter	SPDX				
L70	Bitstream-Vera	SPDX				
L71	bittorrent					GNU
L72	BitTorrent-1.0	SPDX				
L73	BitTorrent-1.1	SPDX		FSF		
L74	blessing	SPDX				
L75	BlueOak-1.0.0	SPDX			OSI	
L76	Boehm-GC	SPDX				
L77	boost					GNU
L78	Borceux	SPDX				
L79	Brian-Gladman-2-Clause	SPDX				
L80	Brian-Gladman-3-Clause	SPDX				
L81	BSD-1-Clause	SPDX		FSF	OSI	
L82	BSD-2-Clause	SPDX		FSF		
L83	BSD-2-Clause-Darwin	SPDX				
L84	BSD-2-Clause-FreeBSD			FSF		
L85	BSD-2-Clause-Patent	SPDX			OSI	
L86	BSD-2-Clause-Views	SPDX				
L87	BSD-3-Clause	SPDX	DFSG	FSF	OSI	
L88	BSD-3-Clause-acpica	SPDX				
L89	BSD-3-Clause-Attribution	SPDX				
L90	BSD-3-Clause-Clear	SPDX		FSF		
L91	BSD-3-Clause-flex	SPDX				
L92	BSD-3-Clause-HP	SPDX				
L93	BSD-3-Clause-LBNL	SPDX			OSI	
L94	BSD-3-Clause-Modification	SPDX				
L95	BSD-3-Clause-No-Military-License	SPDX				
L96	BSD-3-Clause-No-Nuclear-License	SPDX				

L97	BSD-3-Clause-No-Nuclear-License-2014	SPDX				
L98	BSD-3-Clause-No-Nuclear-Warranty	SPDX				
L99	BSD-3-Clause-Open-MPI	SPDX				
L100	BSD-3-Clause-Sun	SPDX				
L101	BSD-4-Clause	SPDX		FSF		
L102	BSD-4-Clause-Shortened	SPDX				
L103	BSD-4-Clause-UC	SPDX				
L104	BSD-4.3RENO	SPDX				
L105	BSD-4.3TAHOE	SPDX				
L106	BSD-Advertising-Acknowledgement	SPDX				
L107	BSD-Attribution-HPND-disclaimer	SPDX				
L108	BSD-Inferno-Nettverk	SPDX				
L109	BSD-Protection	SPDX				
L110	BSD-Source-beginning-file	SPDX				
L111	BSD-Source-Code	SPDX				
L112	BSD-Systemics	SPDX				
L113	BSD-Systemics-W3Works	SPDX				
L114	BSL-1.0	SPDX		FSF	OSI	
L115	BUSL-1.1	SPDX				
L116	bzip2-1.0.6	SPDX				
L117	C-UDA-1.0	SPDX				
L118	CAL-1.0	SPDX			OSI	
L119	CAL-1.0-Combined-Work-Exception	SPDX				
L120	Caldera	SPDX				
L121	Caldera-no-preamble	SPDX				
L122	CATOSL-1.1	SPDX				
L123	CC-BY-1.0	SPDX	DFSG			
L124	CC-BY-2.0	SPDX		FSF		
L125	CC-BY-2.5	SPDX				
L126	CC-BY-2.5-AU	SPDX				
L127	CC-BY-3.0	SPDX	DFSG	FSF		
L128	CC-BY-3.0-AT	SPDX				
L129	CC-BY-3.0-AU	SPDX				
L130	CC-BY-3.0-DE	SPDX				

L131	CC-BY-3.0-IGO	SPDX				
L132	CC-BY-3.0-NL	SPDX				
L133	CC-BY-3.0-US	SPDX				
L134	CC-BY-4.0	SPDX	DFSG	FSF		
L135	CC-BY-NC-1.0	SPDX				
L136	CC-BY-NC-2.0	SPDX				
L137	CC-BY-NC-2.5	SPDX				
L138	CC-BY-NC-3.0	SPDX				
L139	CC-BY-NC-3.0-DE	SPDX				
L140	CC-BY-NC-4.0	SPDX				
L141	CC-BY-NC-ND-1.0	SPDX				
L142	CC-BY-NC-ND-2.0	SPDX				
L143	CC-BY-NC-ND-2.5	SPDX				
L144	CC-BY-NC-ND-3.0	SPDX				
L145	CC-BY-NC-ND-3.0-DE	SPDX				
L146	CC-BY-NC-ND-3.0-IGO	SPDX				
L147	CC-BY-NC-ND-4.0	SPDX				
L148	CC-BY-NC-SA-1.0	SPDX	DFSG			
L149	CC-BY-NC-SA-2.0	SPDX				
L150	CC-BY-NC-SA-2.0-DE	SPDX				
L151	CC-BY-NC-SA-2.0-FR	SPDX				
L152	CC-BY-NC-SA-2.0-UK	SPDX				
L153	CC-BY-NC-SA-2.5	SPDX				
L154	CC-BY-NC-SA-3.0	SPDX				
L155	CC-BY-NC-SA-3.0-DE	SPDX				
L156	CC-BY-NC-SA-3.0-IGO	SPDX				
L157	CC-BY-NC-SA-4.0	SPDX				
L158	CC-BY-ND-1.0	SPDX				
L159	CC-BY-ND-2.0	SPDX				
L160	CC-BY-ND-2.5	SPDX				
L161	CC-BY-ND-3.0	SPDX		FSF		
L162	CC-BY-ND-3.0-DE	SPDX				
L163	CC-BY-ND-4.0	SPDX				
L164	CC-BY-SA-1.0	SPDX	DFSG			
L165	CC-BY-SA-2.0	SPDX		FSF		

L166	CC-BY-SA-2.0-UK	SPDX				
L167	CC-BY-SA-2.1-JP	SPDX				
L168	CC-BY-SA-2.5	SPDX		FSF		
L169	CC-BY-SA-3.0	SPDX	DFSG	FSF		
L170	CC-BY-SA-3.0-AT	SPDX				
L171	CC-BY-SA-3.0-DE	SPDX				
L172	CC-BY-SA-3.0-IGO	SPDX				
L173	CC-BY-SA-4.0	SPDX	DFSG	FSF		
L174	CC-PDDC	SPDX				
L175	CC-SAMPLING+1.0		DFSG			
L176	CC0			FSF		
L177	CC0-1.0	SPDX				
L178	CDDL					GNU
L179	CDDL-1.0	SPDX		FSF		
L180	CDDL-1.1	SPDX				
L181	CDL-1.0	SPDX				
L182	CDLA-Permissive-1.0	SPDX				
L183	CDLA-Permissive-2.0	SPDX				
L184	CDLA-Sharing-1.0	SPDX				
L185	CeCILL					GNU
L186	CECILL-1.0	SPDX				
L187	CECILL-1.1	SPDX				
L188	CECILL-2.0	SPDX		FSF		
L189	CECILL-2.1	SPDX			OSI	
L190	CECILL-B	SPDX				
L191	Cecill-B-v1			FSF		
L192	CECILL-C	SPDX				
L193	Cecill-C-v1			FSF		
L194	CERN-OHL-1.1	SPDX				
L195	CERN-OHL-1.2	SPDX				
L196	CERN-OHL-P-2.0	SPDX			OSI	
L197	CERN-OHL-S-2.0	SPDX			OSI	
L198	CERN-OHL-W-2.0	SPDX			OSI	
L199	CFITSIO	SPDX				
L200	check-cvs	SPDX				

L201	checkmk	SPDX				
L202	ClarifiedArtistic					GNU
L203	ClArtistic	SPDX		FSF		
L204	clearbsd					GNU
L205	Clips	SPDX				
L206	CMU-Mach	SPDX				
L207	CMU-Mach-nodoc	SPDX				
L208	CNRI			FSF		
L209	CNRI-Jython	SPDX				
L210	CNRI-Python	SPDX			OSI	
L211	CNRI-Python-GPL-Compatible	SPDX				
L212	COIL-1.0	SPDX				
L213	comclause					GNU
L214	CommonPublicLicense10					GNU
L215	Commons-Clause			FSF		
L216	Community-Spec-1.0	SPDX				
L217	Condor					GNU
L218	Condor-1.1	SPDX		FSF		
L219	copyleft-next-0.3.0	SPDX				
L220	copyleft-next-0.3.1	SPDX				
L221	CorkforkPL			FSF		
L222	Cornell-Lossless-JPEG	SPDX				
L223	CPAL					GNU
L224	CPAL-1.0	SPDX	DFSG	FSF	OSI	
L225	CPL-1.0	SPDX	DFSG	FSF	OSI	
L226	cpol					GNU
L227	CPOL-1.02	SPDX		FSF		
L228	Cronyx	SPDX				
L229	Crossword	SPDX				
L230	CryptixGeneralLicense					GNU
L231	CryptixGL			FSF		
L232	CrystalStacker	SPDX				
L233	CUA-OPL-1.0	SPDX				
L234	Cube	SPDX				
L235	curl	SPDX		FSF		

L236	cvw				OSI	
L237	D-FSL-1.0	SPDX				
L238	DBG-3.0			FSF		
L239	DEC-3-Clause	SPDX				
L240	DejaVu			FSF		
L241	Design-Science-L			FSF		
L242	diffmark	SPDX				
L243	DL-DE-BY-2.0	SPDX				
L244	DL-DE-ZERO-2.0	SPDX				
L245	DOC	SPDX				
L246	DOR					GNU
L247	Dotseqn	SPDX				
L248	DRL-1.0	SPDX				
L249	DRL-1.1	SPDX				
L250	DSDP	SPDX				
L251	dtoa	SPDX				
L252	dvipdfm	SPDX				
L253	ECL-1.0	SPDX				
L254	ECL-2.0	SPDX		FSF	OSI	
L255	ECos-2.0			FSF	OSI	
L256	eCos11					GNU
L257	EFL-1.0	SPDX			OSI	
L258	EFL-2.0	SPDX		FSF	OSI	
L259	eGenix	SPDX				
L260	Eiffel					GNU
L261	Elastic-2.0	SPDX				
L262	Entessa	SPDX				
L263	EPICS	SPDX		FSF		
L264	EPL					GNU
L265	EPL-1.0	SPDX	DFSG	FSF	OSI	
L266	EPL-2.0	SPDX		FSF	OSI	
L267	EPL2					GNU
L268	ErlPL-1.1	SPDX		FSF		
L269	etalab-2.0	SPDX				
L270	EUDatagrid	SPDX		FSF	OSI	GNU

L271	EUPL-1.0	SPDX				
L272	EUPL-1.1	SPDX		FSF	OSI	
L273	EUPL-1.2	SPDX			OSI	
L274	Eurosym	SPDX				
L275	Expat			FSF		GNU
L276	EZ-Publish-Professional			FSF		
L277	Fair	SPDX				
L278	FAL		DFSG			
L279	FBM	SPDX				
L280	fdk					GNU
L281	FDK-AAC	SPDX				
L282	Ferguson-Twofish	SPDX				
L283	Frameworx-1.0	SPDX				
L284	FreeBSD					GNU
L285	FreeBSD-DL			FSF		
L286	FreeBSD-DOC	SPDX				
L287	FreeImage	SPDX				
L288	Freely-Redistributable			FSF		
L289	freetype					GNU
L290	FSFAP	SPDX		FSF		
L291	FSFAP-no-warranty-disclaimer	SPDX				
L292	FSFUL	SPDX				
L293	FSFULLR	SPDX				
L294	FSFULLRWD	SPDX				
L295	FTL	SPDX		FSF		
L296	Furuseth	SPDX				
L297	fwlw	SPDX				
L298	GCR-docs	SPDX				
L299	GD	SPDX				
L300	GFDL-1.1-invariants-only	SPDX				
L301	GFDL-1.1-invariants-or-later	SPDX				
L302	GFDL-1.1-no-invariants-only	SPDX				
L303	GFDL-1.1-no-invariants-or-later	SPDX				
L304	GFDL-1.1-only	SPDX		FSF		
L305	GFDL-1.1-or-later	SPDX		FSF		

L306	GFDL-1.2-invariants-only	SPDX				
L307	GFDL-1.2-invariants-or-later	SPDX				
L308	GFDL-1.2-no-invariants-only	SPDX				
L309	GFDL-1.2-no-invariants-or-later	SPDX				
L310	GFDL-1.2-only	SPDX		FSF		
L311	GFDL-1.2-or-later	SPDX		FSF		
L312	GFDL-1.3		DFSG			
L313	GFDL-1.3-invariants-only	SPDX				
L314	GFDL-1.3-invariants-or-later	SPDX				
L315	GFDL-1.3-no-invariants-only	SPDX				
L316	GFDL-1.3-no-invariants-or-later	SPDX				
L317	GFDL-1.3-only	SPDX		FSF		
L318	GFDL-1.3-or-later	SPDX		FSF		
L319	Giftware	SPDX				
L320	GL2PS	SPDX		FSF		
L321	Glide	SPDX				
L322	Glulxe	SPDX				
L323	GLWTPL	SPDX				
L324	GNU-Verbatim-C-L			FSF		
L325	GNUAllPermissive					GNU
L326	GNUGPL					GNU
L327	GNUGPLv3					GNU
L328	gnuplot	SPDX		FSF		GNU
L329	GPL-1.0-only	SPDX		FSF		
L330	GPL-1.0-or-later	SPDX		FSF		
L331	GPL-2.0-only	SPDX		FSF		
L332	GPL-2.0-or-later	SPDX		FSF		
L333	GPL-3.0-only	SPDX	DFSG	FSF	OSI	
L334	GPL-3.0-or-later	SPDX		FSF		
L335	GPL-PA			FSF		
L336	GPLv2					GNU
L337	Graphics-Gems	SPDX				
L338	gSOAP-1.3b	SPDX				
L339	gtkbook	SPDX				
L340	HaskellReport	SPDX				

L341	hdparm	SPDX				
L342	HESSLA			FSF		GNU
L343	Hipergate			FSF		
L344	hippocratic					GNU
L345	Hippocratic-2.1	SPDX				
L346	HP-1986	SPDX				
L347	HP-1989	SPDX				
L348	HPND	SPDX		FSF		GNU
L349	HPND-DEC	SPDX				
L350	HPND-doc	SPDX				
L351	HPND-doc-sell	SPDX				
L352	HPND-export-US	SPDX				
L353	HPND-export-US-modify	SPDX				
L354	HPND-Fenneberg-Livingston	SPDX				
L355	HPND-INRIA-IMAG	SPDX				
L356	HPND-Kevlin-Henney	SPDX				
L357	HPND-Markus-Kuhn	SPDX				
L358	HPND-MIT-disclaimer	SPDX				
L359	HPND-Pbmplus	SPDX				
L360	HPND-sell-MIT-disclaimer-xserver	SPDX				
L361	HPND-sell-regexpr	SPDX				
L362	HPND-sell-variant	SPDX				
L363	HPND-sell-variant-MIT-disclaimer	SPDX				
L364	HPND-UC	SPDX				
L365	HTMLTIDY	SPDX				
L366	IBM			FSF		
L367	IBM-pibs	SPDX				
L368	IBMPL					GNU
L369	ICU	SPDX			OSI	
L370	IEC-Code-Components-EULA	SPDX				
L371	IJG	SPDX		FSF		GNU
L372	IJG-short	SPDX				
L373	ImageMagick	SPDX				
L374	iMatix	SPDX		FSF		GNU
L375	imlib					GNU

L376	Imlib2	SPDX		FSF		
L377	Info-ZIP	SPDX		FSF		
L378	informal					GNU
L379	Inner-Net-2.0	SPDX				
L380	Intel	SPDX		FSF		GNU
L381	Intel-ACPI	SPDX		FSF		
L382	Interbase-1.0	SPDX				
L383	IPA	SPDX		FSF	OSI	
L384	IPL-1.0	SPDX	DFSG	FSF		
L385	ISC	SPDX	DFSG	FSF	OSI	GNU
L386	ISC-Veillard	SPDX				
L387	Jahia					GNU
L388	JahiaCSL			FSF		
L389	Jam	SPDX			OSI	
L390	JasPer-2.0	SPDX				
L391	josl					GNU
L392	JOSL-1.0			FSF		
L393	JPL-image	SPDX				
L394	JPNIC	SPDX				
L395	JSON	SPDX	DFSG	FSF		GNU
L396	Kastrup	SPDX				
L397	Kazlib	SPDX				
L398	Knuth-CTAN	SPDX				
L399	ksh93					GNU
L400	LAL-1.2	SPDX				
L401	LAL-1.3	SPDX		FSF		
L402	LaTeX ecfonts			FSF		
L403	Latex2e	SPDX				
L404	Latex2e-translated-notice	SPDX				
L405	Leptonica	SPDX				
L406	LGPL					GNU
L407	LGPL-2.0-only	SPDX		FSF	OSI	
L408	LGPL-2.0-or-later	SPDX		FSF		
L409	LGPL-2.1-only	SPDX		FSF		
L410	LGPL-2.1-or-later	SPDX		FSF		

L411	LGPL-3.0-only	SPDX	DFSG	FSF	OSI	
L412	LGPL-3.0-or-later	SPDX		FSF		
L413	LGPLLR	SPDX		FSF		
L414	LGPLv3					GNU
L415	Lha			FSF		GNU
L416	Libpng	SPDX				
L417	libpng-2.0	SPDX				
L418	libselinux-1.0	SPDX				
L419	libtiff	SPDX				
L420	libutil-David-Nugent	SPDX				
L421	LiLiQ-P-1.1	SPDX			OSI	
L422	LiLiQ-R-1.1	SPDX			OSI	
L423	LiLiQ-Rplus-1.1	SPDX			OSI	
L424	LinkGrammarLicense			FSF		
L425	Linux-man-pages-1-para	SPDX				
L426	Linux-man-pages-copyleft	SPDX				
L427	Linux-man-pages-copyleft-2-para	SPDX				
L428	Linux-man-pages-copyleft-var	SPDX				
L429	Linux-OpenIB	SPDX				
L430	LLGPL			FSF		
L431	LOOP	SPDX				
L432	LPD-document	SPDX				
L433	LPL-1.0	SPDX				
L434	LPL-1.02	SPDX		FSF		
L435	LPPL-1.0	SPDX				
L436	LPPL-1.1	SPDX				
L437	LPPL-1.2	SPDX		FSF		
L438	LPPL-1.3a	SPDX		FSF		
L439	LPPL-1.3c	SPDX		FSF	OSI	
L440	lsof	SPDX				
L441	Lua license			FSF		
L442	lucent102					GNU
L443	Lucida-Bitmap-Fonts	SPDX				
L444	LZMA-SDK-9.11-to-9.20	SPDX				
L445	LZMA-SDK-9.22	SPDX				

L446	Mackerras-3-Clause	SPDX				
L447	Mackerras-3-Clause-acknowledgment	SPDX				
L448	magaz	SPDX				
L449	mailprio	SPDX				
L450	MakeIndex	SPDX				
L451	Martin-Birgmeier	SPDX				
L452	McPhee-slideshow	SPDX				
L453	metamail	SPDX				
L454	Minpack	SPDX				
L455	MirOS	SPDX	DFSG	FSF	OSI	
L456	MIT	SPDX	DFSG		OSI	
L457	MIT-0	SPDX			OSI	
L458	MIT-advertising	SPDX				
L459	MIT-CMU	SPDX				
L460	MIT-enna	SPDX				
L461	MIT-feh	SPDX				
L462	MIT-Festival	SPDX				
L463	MIT-Modern-Variant	SPDX				
L464	MIT-open-group	SPDX				
L465	MIT-testregex	SPDX				
L466	MIT-Wu	SPDX				
L467	MITNFA	SPDX				
L468	MMIXware	SPDX				
L469	Modified X11			FSF		
L470	ModifiedBSD					GNU
L471	Motosoto	SPDX			OSI	
L472	MPEG-SSG	SPDX				
L473	mpi-permissive	SPDX				
L474	mpich2	SPDX				
L475	MPL			FSF		GNU
L476	MPL-1.0	SPDX			OSI	
L477	MPL-1.1	SPDX		FSF	OSI	
L478	MPL-2.0	SPDX	DFSG	FSF	OSI	
L479	MPL-2.0-no-copyleft-exception	SPDX				
L480	mplus	SPDX				

L481	MS-LPL	SPDX				
L482	MS-PL	SPDX		FSF	OSI	
L483	MS-RL	SPDX		FSF	OSI	
L484	Ms-SS			FSF		
L485	MTLL	SPDX				
L486	MulanPSL-1.0	SPDX				
L487	MulanPSL-2.0	SPDX			OSI	
L488	Multics	SPDX			OSI	
L489	Mup	SPDX				
L490	NAIST-2003	SPDX				
L491	NASA					GNU
L492	NASA-1.3	SPDX		FSF		
L493	Naumen	SPDX				
L494	NBPL-1.0	SPDX				
L495	NCGL-UK-2.0	SPDX				
L496	NCSA	SPDX		FSF	OSI	GNU
L497	Net-SNMP	SPDX				
L498	NetCDF	SPDX				
L499	NetscapeJavaScript					GNU
L500	newOpenLDAP					GNU
L501	Newsletr	SPDX				
L502	NGPL	SPDX		FSF	OSI	
L503	NICTA-1.0	SPDX				
L504	NikoSoft Group Public License			FSF		
L505	NIST-PD	SPDX				
L506	NIST-PD-fallback	SPDX				
L507	NIST-Software	SPDX				
L508	NLOD-1.0	SPDX				
L509	NLOD-2.0	SPDX				
L510	NLPL	SPDX				
L511	Nokia	SPDX		FSF	OSI	GNU
L512	NoLicense					GNU
L513	NOSL	SPDX		FSF		GNU
L514	Noweb	SPDX				
L515	NPL					GNU

L516	NPL-1.0	SPDX				
L517	NPL-1.1	SPDX		FSF		
L518	NPOSL-3.0	SPDX			OSI	
L519	NRL	SPDX				
L520	NTP	SPDX			OSI	
L521	NTP-0	SPDX				
L522	O-UDA-1.0	SPDX				
L523	OCCT-PL	SPDX				
L524	OCL-1.0			FSF		
L525	OCLC-2.0	SPDX				
L526	oclc2-php				OSI	
L527	Oculus VR Rift SDK License			FSF		
L528	OculusRiftSDK					GNU
L529	ODbL-1.0	SPDX		FSF		
L530	ODC-By-1.0	SPDX				
L531	OFFIS	SPDX				
L532	OFL-1.0	SPDX				
L533	OFL-1.0-no-RFN	SPDX				
L534	OFL-1.0-RFN	SPDX				
L535	OFL-1.1	SPDX	DFSG	FSF	OSI	
L536	OFL-1.1-no-RFN	SPDX				
L537	OFL-1.1-RFN	SPDX				
L538	OGC-1.0	SPDX				
L539	OGDL-Taiwan-1.0	SPDX				
L540	OGL-Canada-2.0	SPDX				
L541	OGL-UK-1.0	SPDX				
L542	OGL-UK-2.0	SPDX				
L543	OGL-UK-3.0	SPDX				
L544	OGTSL	SPDX			OSI	
L545	Old ksh93			FSF		
L546	Old-plan9			FSF		
L547	OLDAP-1.1	SPDX				
L548	OLDAP-1.2	SPDX				
L549	OLDAP-1.3	SPDX				
L550	OLDAP-1.4	SPDX				

L551	OLDAP-2.0	SPDX				
L552	OLDAP-2.0.1	SPDX				
L553	OLDAP-2.1	SPDX				
L554	OLDAP-2.2	SPDX				
L555	OLDAP-2.2.1	SPDX				
L556	OLDAP-2.2.2	SPDX				
L557	OLDAP-2.3	SPDX		FSF		
L558	OLDAP-2.4	SPDX				
L559	OLDAP-2.5	SPDX				
L560	OLDAP-2.6	SPDX				
L561	OLDAP-2.7	SPDX		FSF		
L562	OLDAP-2.8	SPDX		FSF	OSI	
L563	oldOpenLDAP					GNU
L564	OLFL-1.3	SPDX			OSI	
L565	OML	SPDX				
L566	Open Publication License v1.0			FSF		
L567	OpenPBS-2.3	SPDX	DFSG			
L568	OpenPublicL					GNU
L569	OpenSSL	SPDX		FSF		GNU
L570	OpenSSL-standalone	SPDX				
L571	OpenVision	SPDX				
L572	OPL-1.0	SPDX	DFSG	FSF		
L573	OPL-UK-3.0	SPDX				
L574	OPUBL-1.0	SPDX				
L575	OriginalBSD					GNU
L576	OSET-PL-2.1	SPDX			OSI	
L577	OSL					GNU
L578	OSL-1.0	SPDX			OSI	
L579	OSL-1.1	SPDX	DFSG			
L580	OSL-2.0	SPDX				
L581	OSL-2.1	SPDX			OSI	
L582	OSL-3.0	SPDX		FSF	OSI	
L583	PADL	SPDX				
L584	Parity-6.0.0	SPDX				
L585	Parity-7.0.0	SPDX				

L586	PCRE			FSF		
L587	PDDL-1.0	SPDX				
L588	Perl			FSF		
L589	PerlLicense					GNU
L590	Phorum					GNU
L591	Phorum-2.0			FSF		
L592	PHP			FSF		
L593	PHP-2.02			FSF		
L594	PHP-3.0	SPDX		FSF	OSI	
L595	PHP-3.01	SPDX		FSF	OSI	
L596	PINE			FSF		GNU
L597	Pixar	SPDX				
L598	Plan9					GNU
L599	Plexus	SPDX				
L600	pnmstitch	SPDX				
L601	PolyForm-Noncommercial-1.0.0	SPDX				
L602	PolyForm-Small-Business-1.0.0	SPDX				
L603	PostgreSQL	SPDX			OSI	
L604	PPL			FSF		GNU
L605	PPL3a					GNU
L606	PSF-2.0	SPDX			OSI	
L607	psfrag	SPDX				
L608	psutils	SPDX				
L609	PublicDomain			FSF		GNU
L610	Python			FSF		GNU
L611	Python-1.6a2			FSF		
L612	Python-1.6b1			FSF		
L613	Python-2.0	SPDX				
L614	Python-2.0.1	SPDX		FSF		
L615	Python-2.1.1			FSF		
L616	Python-2.3			FSF		
L617	Python-2.5			FSF		
L618	python-ldap	SPDX				
L619	PythonOld					GNU
L620	Qhull	SPDX				

L621	QPL					GNU
L622	QPL-1.0	SPDX	DFSG	FSF	OSI	
L623	QPL-1.0-INRIA-2004	SPDX				
L624	radvd	SPDX				
L625	Rdisc	SPDX				
L626	Review:EJBCA-REV-ID-2			FSF		
L627	RHeCos-1.1	SPDX		FSF		
L628	RPL					GNU
L629	RPL-1.1	SPDX			OSI	
L630	RPL-1.3			FSF		
L631	RPL-1.5	SPDX			OSI	
L632	RPSL					GNU
L633	RPSL-1.0	SPDX	DFSG	FSF	OSI	
L634	RSA-MD	SPDX				
L635	RSCPL	SPDX			OSI	
L636	Ruby	SPDX		FSF		GNU
L637	SAX-PD	SPDX				
L638	SAX-PD-2.0	SPDX				
L639	Saxpath	SPDX				
L640	SCEA	SPDX				
L641	SchemeReport	SPDX				
L642	Scilab					GNU
L643	Scilab-old			FSF		
L644	SCOSL-3.0			FSF		
L645	Scratch			FSF		GNU
L646	SCSL-2.8			FSF		
L647	Sendmail	SPDX		FSF		
L648	Sendmail-8.23	SPDX				
L649	SGI-B-1.0	SPDX				
L650	SGI-B-1.1	SPDX				
L651	SGI-B-2.0	SPDX		FSF		
L652	SGI-OpenGL	SPDX				
L653	SGIFreeB					GNU
L654	SGP4	SPDX				
L655	SHL-0.5	SPDX				

L656	SHL-0.51	SPDX				
L657	SimPL-2.0	SPDX			OSI	
L658	Simple Permissive With Rules Modified Versions			FSF		
L659	SimpleM			FSF		
L660	SimplePermissive			FSF		
L661	SimplePermissiveNoNonWarranty			FSF		
L662	SimplePermissiveNonWarranty			FSF		
L663	SIP			FSF		
L664	SISSL	SPDX		FSF	OSI	GNU
L665	SISSL-1.2	SPDX		FSF		
L666	SL	SPDX				
L667	Sleepycat	SPDX		FSF	OSI	
L668	SML					GNU
L669	SMLNJ	SPDX		FSF		
L670	SMPPL	SPDX				
L671	SNIA	SPDX				
L672	snprintf	SPDX				
L673	softSurfer	SPDX				
L674	Soundex	SPDX				
L675	Spencer-86	SPDX		FSF		
L676	Spencer-94	SPDX				
L677	Spencer-99	SPDX				
L678	spin		DFSG			
L679	SPL					GNU
L680	SPL-1.0	SPDX		FSF	OSI	
L681	Squeak					GNU
L682	Squeak-old			FSF		
L683	ssh-keyscan	SPDX				
L684	SSH-OpenSSH	SPDX				
L685	SSH-short	SPDX				
L686	SSLeay-standalone	SPDX				
L687	SSPL-1.0	SPDX				
L688	SSSCFR-1.1			FSF		
L689	StandardMLofNJ					GNU

L690	SugarCRM-1.1.3	SPDX				
L691	Sun-PPP	SPDX				
L692	SunCommunitySourceLicense					GNU
L693	SunPro	SPDX				
L694	SunSolarisSourceCode					GNU
L695	swisseph		DFSG			
L696	SWL	SPDX				
L697	swrule	SPDX				
L698	Symlinks	SPDX				
L699	TAPR-OHL-1.0	SPDX				
L700	TCL	SPDX		FSF		
L701	TCP-wrappers	SPDX				
L702	TermReadKey	SPDX				
L703	TGPPL-1.0	SPDX		FSF		
L704	THL-1.1			FSF		
L705	TMate	SPDX				
L706	TORQUE-1.1	SPDX				
L707	TOSL	SPDX				
L708	TPDL	SPDX				
L709	TPL-1.0	SPDX				
L710	TrueCrypt			FSF		
L711	TTWL	SPDX				
L712	TTYP0	SPDX				
L713	TU-Berlin-1.0	SPDX				
L714	TU-Berlin-2.0	SPDX				
L715	UCAR	SPDX				
L716	UCL-1.0	SPDX			OSI	
L717	ulem	SPDX				
L718	UMich-Merit	SPDX				
L719	Unicode					GNU
L720	Unicode-3.0	SPDX				
L721	Unicode-DFS-2012			FSF		
L722	Unicode-DFS-2015	SPDX			OSI	
L723	Unicode-DFS-2016	SPDX				
L724	Unicode-TOU	SPDX				

L725	UnixCrypt	SPDX				
L726	Unlicense	SPDX			OSI	GNU
L727	UPL					GNU
L728	UPL-1.0	SPDX			OSI	
L729	URT-RLE	SPDX				
L730	UtahPublicLicense					GNU
L731	Vim	SPDX				GNU
L732	VOSTROM	SPDX				
L733	VSL-0.1				OSI	
L734	VSL-1.0	SPDX				
L735	W3C	SPDX				GNU
L736	W3C-19980720	SPDX				
L737	W3C-20150513	SPDX			OSI	
L738	w3m	SPDX				
L739	Watcom					GNU
L740	Watcom-1.0	SPDX				
L741	WebM					GNU
L742	Widget-Workshop	SPDX				
L743	Wsuipa	SPDX				
L744	WTFPL	SPDX	DFSG			GNU
L745	Wx					GNU
L746	Wxwind					GNU
L747	wxWindows				OSI	
L748	x-oz		DFSG			
L749	X11	SPDX				
L750	X11-distribute-modifications-variant	SPDX				
L751	X11License					GNU
L752	Xdebug-1.03	SPDX				
L753	Xerox	SPDX				
L754	Xfig	SPDX				
L755	XFree86-1.1	SPDX				
L756	xinetd	SPDX				GNU
L757	xkeyboard-config-Zinoviev	SPDX				
L758	xlock	SPDX				
L759	Xnet	SPDX			OSI	

L760	xpp	SPDX				
L761	XSkat	SPDX				
L762	Yahoo					GNU
L763	YaST					GNU
L764	YPL-1.0	SPDX				
L765	YPL-1.1	SPDX				
L766	Zed	SPDX				
L767	Zeeff	SPDX				
L768	Zend					GNU
L769	Zend-2.0	SPDX				
L770	Zimbra					GNU
L771	Zimbra-1.3	SPDX				
L772	Zimbra-1.4	SPDX				
L773	Zlib	SPDX	DFSG		OSI	GNU
L774	zlib-acknowledgement	SPDX				
L775	Zope					GNU
L776	ZPL - 2.0				OSI	
L777	ZPL - 2.1				OSI	
L778	ZPL-1.1	SPDX				
L779	ZPL-2.0	SPDX				
L780	ZPL-2.1	SPDX				

Appendix B Primary literature reviewed, read in full and inclusion/exclusion criteria applied

Table B.1: List of literature with the inclusion/exclusion criteria applied.

Literature identifier	Identifier	SPDX	DFSG	FSF	OSI	GNU
L1	0BSD	SPDX			OSI	
L2	996			FSF		
L3	AAL	SPDX			OSI	
L4	Abstyles	SPDX				
L5	AcademicFreeLicense					GNU
L6	ACDL-1.0			FSF		
L7	ACEL			FSF		
L8	AdaCore-doc	SPDX				
L9	Adobe-2006	SPDX				
L10	Adobe-Display-PostScript	SPDX				
L11	Adobe-Glyph	SPDX				

Appendix C Primary literature reviewed, read in full and data extracted

Table C.1: Final list of literature and their data extraction

Literature identifier	Identifier	SPDX	DFSG	FSF	OSI	GNU
L1	0BSD	SPDX			OSI	
L2	996			FSF		
L3	AAL	SPDX			OSI	
L4	Abstyles	SPDX				
L5	AcademicFreeLicense					GNU
L6	ACDL-1.0			FSF		
L7	ACEL			FSF		
L8	AdaCore-doc	SPDX				
L9	Adobe-2006	SPDX				
L10	Adobe-Display-PostScript	SPDX				
L11	Adobe-Glyph	SPDX				