



Master's thesis

Master's Programme in Computer Science

# **PCLs in Software Engineering: A Systematic Multivocal Literature Review**

Akira Taguchi

March 4, 2024

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](mailto: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		
PCLs in Software Engineering: A Systematic 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	March 4, 2024	11 pages, 3 appendix pages
Tiivistelmä — Referat — Abstract		
<p>Public copyright licenses (PCL) play a major part 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. sub-problem here. thesis' contribution to solution here.</p> <p>Tell about the research method here.</p> <p>Tell about the results here.</p> <p>Tell about the discussion here.</p>		
<p><b>ACM Computing Classification System (CCS)</b>          Social and professional topics → Computing / technology policy → Intellectual property → Licensing</p>		
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

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Research goal, questions and contributions . . . . .	2
1.2	Thesis structure . . . . .	3
1.3	Background and terminology of PCLs . . . . .	3
<b>2</b>	<b>Methods</b>	<b>7</b>
2.1	Research questions . . . . .	7
2.2	Search stragegy . . . . .	7
2.2.1	Search method . . . . .	7
2.2.2	Search scope and terms . . . . .	7
2.3	Search process . . . . .	7
2.4	Inclusion and exclusion criteria . . . . .	7
2.5	Quality and evidence criteria . . . . .	7
2.6	Data collection and data analysis . . . . .	7
<b>3</b>	<b>Results</b>	<b>8</b>
3.1	Placeholder question (RQ1) . . . . .	8
3.2	Placeholder question (RQ2) . . . . .	8
3.3	Placeholder qiestion (RQ3) . . . . .	8
<b>4</b>	<b>Discussion</b>	<b>9</b>
4.1	Implications for research . . . . .	9
4.2	Implications for software engineering professionals . . . . .	9
4.3	Limitations and threats to validity . . . . .	9
4.3.1	Limitations of license selection for review . . . . .	9
4.3.2	Limitations in data extraction . . . . .	9
<b>5</b>	<b>Conclusions</b>	<b>10</b>
5.1	Future research . . . . .	10

- A Primary studies identified in the search process and their inclusion/exclusion criteria
- B Primary studies reviewed in the quality criteria step and manual exclusions reasons
- C Primary studies reviewed, read in full and data extracted

# 1 Introduction

PCLs play a major part 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. Because open source is central to software engineering the licenses enabling open source must also be considered important in the same context.

Public copyright license is defined by Wikipedia with the following words (Category:PCLs, 2012):

”A PCL 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 correctness 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 multivocal literature through a systematic 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 secondary goal of this research is to conduct a systematic multivocal literature review of the current state of the 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 often do PCLs in software engineering change?
- RQ2: How have the PCLs in software engineering changed?
- RQ3: How long is the average PCL in software engineering?
- RQ4: What are the common reasons for version changes 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 terminology and definitions and will be used to establish a sound basis for discussing this broad subject.

This study has two goals: 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 licensors would make the correct choices based on their situations and needs in a mindful way.

## 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 systematic multivocal literature review. Biggest part of the research happens here. 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 includes also the PCLs that are not approved by the FSF nor OSI hence not falling under the group of FLOSS licenses. 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).

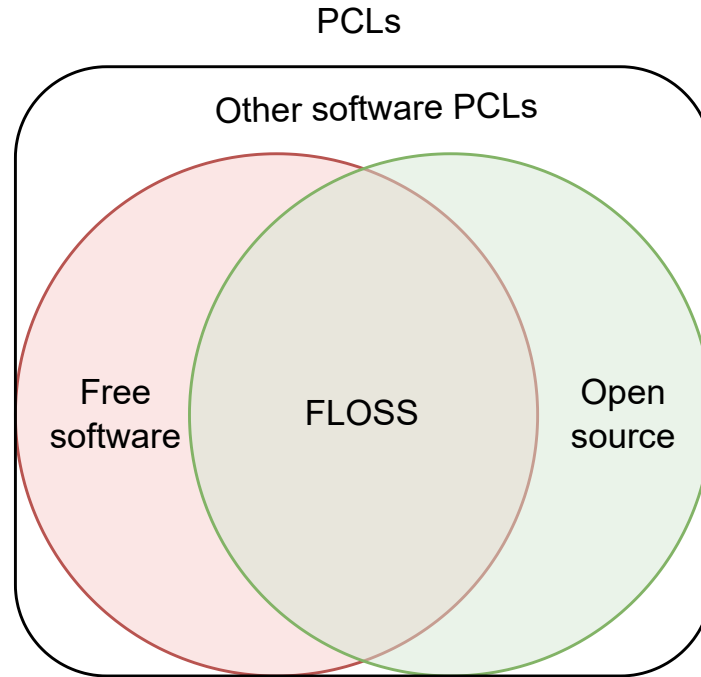
These domains include:

- 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 the 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 the PCLs has seen discussion in the academic field of law since the dawn of the 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).



**Figure 1.1:** PCLS in software engineering

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.

grammar specifications

figure from grammar

define PCLS in se

the connection in terms figure

acknowledge the topic is complex

## **2 Methods**

### **2.1 Research questions**

### **2.2 Search stragey**

#### **2.2.1 Search method**

#### **2.2.2 Search scope and terms**

### **2.3 Search process**

### **2.4 Inclusion and exclusion criteria**

### **2.5 Quality and evidence criteria**

### **2.6 Data collection and data analysis**

## 3 Results

3.1 Placeholder question (RQ1)

3.2 Placeholder question (RQ2)

3.3 Placeholder question (RQ3)

## 4 Discussion

### 4.1 Implications for research

### 4.2 Implications for software engineering professionals

### 4.3 Limitations and threats to validity

#### 4.3.1 Limitations of license selection for review

#### 4.3.2 Limitations in data extraction

# 5 Conclusions

## 5.1 Future research



# Bibliography

- Category:PCLs (2012). *Category:PCLs* — Wikipedia, The Free Encyclopedia. [https://web.archive.org/web/20240131085240/https://en.wikipedia.org/wiki/Category:Public\\_copyright\\_licenses](https://web.archive.org/web/20240131085240/https://en.wikipedia.org/wiki/Category:Public_copyright_licenses). Accessed: 2024 Jan 31.
- 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>.
- 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>.
- 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.
- 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.
- 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.
- OSI (2008). *GNU General Public License version 3*. <https://web.archive.org/web/20240226075409/https://opensource.org/licenses/gpl-3-0>. Accessed: 2024 Feb 26.
- 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](https://doi.org/10.1145/1516046.1516058). URL: <https://doi.org/10.1145/1516046.1516058>.



## **Appendix A Primary studies identified in the search process and their inclusion/exclusion criteria**

## **Appendix B Primary studies reviewed in the quality criteria step and manual exclusions reasons**

## Appendix C Primary studies reviewed, read in full and data extracted