

A Guided Tour of the Legal Implications of Software Cloning

Paolo Ciancarini
University of Bologna & CINI
Italy
paolo.ciancarini@unibo.it

Daniel Russo^{*}
University of Bologna & CINI
Italy
daniel.russo@unibo.it

Alberto Sillitti
Center for Applied Software
Engineering & CINI
Italy
alberto@case-research.it

Giancarlo Succi
Innopolis University
Kazan, Russian Federation
giancarlo.succi@innopolis.ru

ABSTRACT

Software Cloning is the typical example where an interdisciplinary approach may bring additional elements into the community's discussion. In fact, little research has been done in its analysis from an Intellectual Propriety Rights (IPRs) perspective, even if it is a widely studied aspect of software engineering. An interdisciplinary approach is crucial to better understand the legal implications of software in the IPR context. Interestingly, the academic community of software and systems deals much more with such IPR issues than courts themselves. In this paper, we analyze some recent legal decisions in using software clones from a software engineering perspective. In particular, we survey the behavior of some major courts about cloning issues. As a major outcome of our research, it seems that legal *fora* do not have major concerns regarding copyright infringements in software cloning. The major contribution of this work is a case by case analysis of more than one hundred judgments by the US courts and the European Court of Justice. We compare the US and European courts case laws and discuss the impact of a recent European ruling. The US and EU contexts are quite different, since in the US software is patentable while in the EU it is not. Hence, European courts look more permissive regarding cloning, since "principles," or "ideas," are not copyrightable by themselves.

CCS Concepts

•Social and professional topics → Copyrights;

^{*}Corresponding author.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ICSE '16 Companion, May 14-22, 2016, Austin, TX, USA

Copyright 2016 ACM 978-1-4503-4205-6/16/05...\$15.00

<http://dx.doi.org/10.1145/2889160.2889220>

Keywords

Software Cloning; Software Reuse; IPR; Copyright

1. INTRODUCTION

A *software clone* is a fragment of source or executable code, that is copied in the same program or in a different one, whereas the act of copy is called *software cloning* [20].

Software cloning is a form of software reuse; in fact clones are identical or similar pieces of code, designs, or other artifacts exploited during the development of a software system. The copy-paste of someone else's code fragments into a different author's software program is a widely used programming practice, ranging from 5%-15% of the code base [24] up to 50% [22]. On average, the reuse of other people's code in large software programs is estimated around 20%-30% of code [1].

There are several ways of reusing code that are more formal (such as software components [7], web services [19], etc.) in which licensing problems are addressed explicitly (e.g., in open source software [18]). However, developers sometimes prefer different and more informal approaches [21]. There are many reasons why programmers copy software fragments and these reasons are largely studied in technical literature [12]. Several authors have already explored a model that studies the interaction and tracking of software licenses. For example, [6] developed a model which describes the interconnection of components from a legal point of view, using document integration patterns that are commonly used to solve the license mismatch problem in practice. For Open Source licenses, [5] proposes an approach to automatically track changes occurring in the licensing terms of a system. However, those reasons are not the focus of this study; what we want to address here are not the technical advantages or disadvantages of cloning software but the behavior of courts. In fact, this work is a study on the main rulings of software cloning from a software engineering perspective, following an approach started in [2], where the focus was on how software patents can influence software designers.

The issue of reuse by cloning is widely studied in Software

Engineering, for instance, [20] lists hundreds of papers; however, in some situations cloning is considered unlawful. In fact, since in several countries, and especially in Europe, software is protected by the copyright law, software cloning is a form of plagiarism. We found this topic particularly relevant from a society’s perspective, since this aspect of software engineering has wide cross effects, well beyond the technical dimension.

However, the definition of plagiarism for software is controversial. For instance, software clones are known to be closely related to various issues in the design of software for games, especially with respect to originality and creativity, qualities that have to be evaluated when an investigation of plagiarism takes place. For instance, in some competitions for software designers, notably in the World Computer Chess Championship, there is an “originality” rule, which requires that all competing programs must either be original or quote other programmers whose work was used. Such a rule has been invoked a number of times, accusing some author of cheating by plagiarizing code to create a program [3]. These discussions about plagiarism are even more intriguing in the case of open source software [27, 23]. “If to plagiarize is to borrow *too much* code, then one needs to decide exactly how much is too much” [3].

Deciding about plagiarism is difficult. Trying to demonstrate that a program has been copied is not simple, for instance there are clones that reproduce only the functionality of a program, while the source code is different.

We did not find in literature a similar research dealing with court’s perspective. We are only aware of a similar paper published in 1996 (by one of the authors of this paper) which outlines some legal implication regarding software reuse in general within the European Union [26]. The main contribution of this work is to survey the case law of these issues, as the court’s point of view.

With this paper we want to offer an insight for researchers and practitioners to understand the ‘way of thinking’ of US and EU courts when dealing with software cloning and, more generally, to IPR issues.

The main considerations are summarized after each section. The structure of this paper is as follows. In Section 2 a brief explanation of the different clone types is given. To understand the main reference points of courts, in Section 3 a brief overview of the main laws are depicted. In Section 4 we carried out all relevant US and European Court of Justice case laws. A manual analysis of both case law was performed. Moreover, an analysis of the European Court of Justice was carried out, to compare both jurisprudential leanings of the courts. We found out that one ruling has a particularly disruptive nature, thus, in Section 5 we discuss it since we believe it will have a huge impact on software copyright in general. In Section 6 we discuss some of the major implication of this paper. Eventually, we conclude and outline some future research in Section 7.

2. BACKGROUND: TYPES OF SOFTWARE CLONES

Software clones are not just copy-paste fragments of differ-

ent codes. Rattan *et al.* [20] identify four types of clones, summarized in Table 1.

Table 1: Clone types in Rattan *et al.* (2013)

Type	Main characteristics
1-Exact Clones	Program fragments which are identical, with the only exception of white lines and comments
2-Parameterized Clones	Program fragments structurally or syntactically similar, with exception for identifiers, literals, types, and layouts
3-Near Miss Clones	Program fragments copied with some modifications in the source code, i.e statements insertions/deletions, besides identifiers, literals, types, and layouts
4-Semantic Clones	Functionally similar program fragments, which are not formally identical

Assessing the lawfulness of the reuse of program fragments is not straightforward. Clone detection tools, even the most reliable ones, are based on heuristic methods, i.e., they make probabilistic judgments, so it is legally impossible to use them to state the existence of cloning, *beyond reasonable doubt*. Since the provision of the reasonable doubt is a fundamental human right, courts have some obvious difficulties in assessing legal responsibilities when a software contains clones of type 4 (functional clones) but also of type 3 (near miss clones).

Since cloning is a complex issue, *a priori* it is neither a good, nor a bad habit to clone software fragments. In Table 2 we summarize some of the main remarks in literature about cloning. This is relevant for the analysis of the case law as it shows the level of maturity of the debate from a software engineering perspective.

3. SHORT COMPARISON OF TWO LEGAL FRAMEWORKS

Courts do not take subjective decisions, they interpret the law. Here, we briefly describe the legal frameworks of the US and the EU regarding software’s IPRs. Both US and EU copyright laws very briefly described, to give a short overview and references about courts’ starting points.

3.1 IPRs in the US

The US law system is based on “Common Law”: this means that previous judgments are binding. Therefore, there are several past cases that are relevant for a court to issue its judgment. Nevertheless, there is a structured codex, where all courts shall take reference to. For copyright cases it is the Title 17 of 1976 and its amendments. The last one is the Reauthorization Act of 2014. Title 17 is composed by 13

Table 2: Advantages and disadvantages of code cloning

Advantages	Disadvantages
Clones are useful if different customers share similar requirements [13]	High maintenance costs [17]
Some programming languages encourage the use of templates, which result in software cloning [13]	Propagation of bugs: if a clone contains an error, it will spread rapidly over other parts of the program [10]
The use of clones can respond, sometimes, to efficiency requirements in the development [11]	Cloning discourages the use of refactoring, leading to a bad design of the system [15]
Using clones reduces the time required to develop a program [11] [12]	Using clones increases the size of the code, leading to a less efficient system [14]

chapters and 14 appendixes. Each chapter regards a specific issue of copyright, as described in the following Table 3.

Appendixes concern integrations with international standards and agreements and amendments of the Copyright Act of 1976. They are composed as detailed in Table 4.

Courts may take into consideration other laws. For instance, anti-monopoly provisions may apply for cases where functional cloning is permitted, to restore market competition. Moreover anti-fraud laws could be claimed in cases where hardware-software is cloned.

In short, even though software cloning is most closely related to copyright infringements, it is not exclusive. Also other laws and provisions could be used by courts to restore justice. Hence, the issue of cloning is more complex and broader than just copyright infringement.

3.2 IPRs in the EU

Similarly, the European Court of Justice (ECJ) has to follow the EU law. Moreover, the ECJ acts like a Supreme Court of all EU Member States. The particularity is that each Member State has its own national law, which is different from the others. However, each ECJ judgment has to be considered binding for each Member State's court. We can consider the EU law system as a mixed civil law and common law system. The most common EU laws regarding copyright are represented in Table 5.

3.3 Consequences for software cloning

The two legal systems deal differently with software cloning, as summarized in Table 6. For our purposes, the differences mean that the concept of “*software cloning*” in the US is quite mature compared to the European one. Since US courts are bound to a common law legal system, their judgments reflect in some sense the *Zeitgeist*, namely the

Table 3: Chapters of Title 17 (Copyright Act)

Chapter	Subject
Chapter 1	Subject Matter and Scope of Copyright
Chapter 2	Copyright Ownership and Transfer
Chapter 3	Duration of Copyright
Chapter 4	Copyright Notice, Deposit, and Registration
Chapter 5	Copyright Infringement and Remedies
Chapter 6	Importation and Exportation
Chapter 7	Copyright Office
Chapter 8	Proceedings by Copyright Royalty Judges
Chapter 9	Protection of Semiconductor Chip Products
Chapter 10	Digital Audio Recording Devices and Media
Chapter 11	Sound Recordings and Music Videos
Chapter 12	Copyright Protection and Management Systems
Chapter 13	Protection of Original Designs

spirit of the time. Such judgments are based on both previous court's decisions and on the judge's interpretation of the case. Therefore, common law judges are much more exposed to the spirit of the time than the EU ones. In fact, several judgments refer directly to such concept, so we can find easily such judgments in the legal databases. Instead, the ECJ is much younger than the US legal system. Since the EU legal system is based on a civil law system, all software cloning judgments have to be explicitly related to the IPR acts. Unfortunately, none EU law treats “software cloning” as such. This means that in the Eur-lex database used to perform our investigation we expect to find most judgments about software cloning under the keywords “*software & copyright*”.

4. THE CASE LAW

4.1 Research Protocol

Adapting a Systematic Literature Review (SLR) for law cases has strong limitations, since case law is not comparable to literature articles. Moreover, articles can be found in the editor's database (e.g., Springer, Elsevier, IEEE, ACM); whereas rulings do not have editors. Rulings are written by judges, after that employees of commercial databases (e.g., LexisNexis) put them on line. Therefore, some rulings are not into any database. Especially rulings issued by lower courts are hardly collected, since they are considered of low relevance.

Moreover, search criteria for a SLR are different from a case law research. In SLR keywords used represents the scientific topic the article addresses. Case law induce a technical keywords, but from the law domain, not software engineering ones. If, with a SLR we have a direct technical feedback about the paper's topic, searching within case law it is always a question of the right interpretation, judgment per

Table 4: Appendixes of Title 17 (Copyright Act)

Appendix	Subject
Appendix A	The Copyright Act of 1976
Appendix B	The Digital Millennium Copyright Act of 1998
Appendix C	The Copyright Royalty and Distribution Reform Act of 2004
Appendix D	The Satellite Home Viewer Extension and Reauthorization Act of 2004
Appendix E	The Intellectual Property Protection and Courts Amendments Act of 2004
Appendix F	The Prioritizing Resources and Organization for Intellectual Property Act of 2008
Appendix G	The Satellite Television Extension and Localism Act of 2010
Appendix H	Title 18 - Crimes and Criminal Procedure, U. S. Code
Appendix I	Title 28 - Judiciary and Judicial Procedure, U. S. Code
Appendix J	Title 44 - Public Printing and Documents, U. S. Code
Appendix K	The Berne Convention Implementation Act of 1988
Appendix L	The Uruguay Round Agreements Act of 1994
Appendix M	GATT Trade-Related Aspects of Intellectual Property Rights (TRIPs) Agreement, Part II
Appendix N	Definition of Berne Convention Work

judgment.

The arbitrary dimension in the analysis of case law is ineluctable. Nevertheless, we elaborate a design for a case law research protocol to systematize our research [4]. To comply as much as possible with the empirical software analysis tradition we modeled our design according to the framework proposed by [25]. We elaborated and followed this protocol:

1. identification of the country/legal system where to carry out the analysis;
2. definition of an appropriate database for the case law;
3. definition of an appropriate query, according to the legal system;
4. manual identification of the relevant cases:
 - (a) exclusion criteria: not relevance to the topic;
 - (b) inclusion criteria: relevance to the topic;
5. manual analysis of the relevant cases.

According to our research protocol, we choose to analyze and compare the US and the EU legal system’s leaning to the cloning issue.

Table 5: EU Directives regarding Copyright

Directive	Subject
Council Directive 93/83/EEC	Copyright and related rights: satellite broadcasting and cable retransmission
Directive 98/84/EC	Protecting electronic pay services against piracy
Directive 96/9/EC	Legal protection: databases
Directive 2001/29/EC	Copyright and related rights in the information society
Directive 2001/84/EC	Resale right for the benefit of the author of an original work of art
Directive 2006/116/EC	Copyright and related rights: term of protection
Directive 2006/115/EC	Rental, lending and certain other rights related to copyright in the field of intellectual property
Directive 2009/24/EC	Legal protection: computer programs
Directive 2012/28/EU	Wider access to copyright material - Orphan works

The database identified for the US is LexisNexis, which is one of the most complete and reliable law databases for the US case law¹. For the EU we used Eur-Lex². Eur-Lex has one big advantage, it is the official Law database of the European Court of Justice. So, any judgment of the ECJ is available in the Eur-Lex database.

The query chosen for for the US was “software & cloning” in the Academic Search for State and Federal Cases of all available judgments of all US courts. The outcome were 85 cases. The identified query for the EU was “software & copyright” in the textual research and the output were 27 cases. Both queries were carried out on July 29th, 2015. As explained before in Subsection 3.3 the two legal systems are different. To respect these diversities we had to adapt our queries, according to our protocol. No EU law refers directly to “software cloning”, but to “software copyright”. Similarly, the US adopt better the *Zeitgeist* and are more confident to speak about software cloning as such.

Both outcomes of 85 cases for the US and 27 for the EU were reproduced in two tables, one for the US³ and one for the EU⁴ with all relevant identifiers. After that, case by case were manually analyzed.

In total, the excluded cases were 51 for the US and 23 for the EU. The inclusion, or exclusion of cases were carried out by the authors on a qualitative basis. However, also the excluded cases were clearly mentioned within the tables, ex-

¹www.lexisnexis.com

²www.eur-lex.europa.eu

³For the US Case Law: www.cs.unibo.it/~cianca/wwwpages/datapapers/US_Case_Law_LexisNexis_After_1944_software_cloning.pdf

⁴For the EU Case Law: www.cs.unibo.it/~cianca/wwwpages/datapapers/EU_Case_Law_EUR-LEX_software_copyright.pdf

Table 6: Main differences between the US and EU legal system concerning this paper

US	EU
Common Law (Judicial decisions are binding – decisions of the highest court can generally only be overturned by that same court or through legislation)	Civil Law (Only legislative enactments are considered binding for all. However, ECJ ruling are considered to be binding in all Member States of the Union)
Long lasting judicial tradition	Recent establishment (1952)
Extensive freedom of contract, courts are more sensitive to the Zeitgeist. Judges may interpret the undergoing issue according to ‘language’ of the counterparties to reestablish the rule of law.	Less freedom of contract and reference to the law. Courts may reestablish the rule of law according to the legislative interpretation on which counterparties are bound

plaining why they were excluded. The motivation of exclusion was the non relevance of the subject matter. The remaining relevant cases were then chosen for the manual analysis.

Relevant cases were analyzed one by one. In total, analyzed cases were 34 for the US and 4 for the EU.

4.2 First considerations about the outcome

The chosen queries were kept as general as possible to catch the highest number of relevant cloning cases. As a consequence, the relevant cases are a minority of the dataset.

However, we are aware of at least 5 more cases, which were not captured by our query. We found them by serendipity, while studying the topic. For completeness we analyzed also these cases.

Moreover, we took just these two systems, because we are most aware of, also due to the language. Definitely it would have been interesting to study also other legal systems regarding cloning issues. Unfortunately, there is a huge language barrier that we have to overcome.

We remark that we only considered case law at EU level, since each Member State has its own case law in its own language.

4.3 The United States

Claims regarding software cloning are rare, probably because they are very difficult to prove. We analyzed all 85 output cases on LexisNexis about “software & cloning” and created the table US Case Law. Interestingly, as shown by our table, the US case law regarding cloning issues in software is wavering and considers mainly copyright issues. There is no single judgment of the Supreme Court about the issue. This means that there is no unequivocal interpretation about cloning, but courts judge, case by case, according to the specific issue (our A, B or C cluster).

We found no case about software clones of types 1, 2, and 3; instead the US courts dealt with functional clones (type 4), as shown in the table. Courts tend to judge over general issues regarding copyright, and do not enter technically in the cloning issue.

Interestingly, courts seem to use common patterns. Therefore we cluster the cases according to these patterns. It was quite surprising to discover that, with regard to software cloning, US courts lead their judgment within one of three clusters. This is probably due to the fact that sticking to the law, the type of evaluation needs a certain level of homogeneity. The choice of putting one case in a cluster were made by the authors, according to the relevance. Since any case is described, the relevance to the cluster may be confirmed also by the reader.

So, to analyze better the Case Law, we clustered the cases into three areas in Table 7. In the next subsections a general description of each cluster will be provided, to motivate our findings.

Table 7: Number of cases per cloning area

Type of Software Cloning	# Cases
A: Software and Hardware cloning issues related to physical devices	13
B: Software cloning issues related to competition and antitrust issues	7
C: Software cloning issues related to misappropriation of trade secrets and copyright infringements	14
NA: Not Applicable. This is not a case related to software cloning	51

According to our research, in the US, we found no relevant case regarding software cloning of types 1, 2, and 3. We do not know if no one has ever claimed a court about the resolution of a cloning issue or courts do not judge over these issues due to their technical complexity. Probably, according to the plaintiff’s strategy, it is better to appeal the court for some more evident issues (like a similar GUI) rather than a piece of code. Courts want to deal, apparently with what they see, like Graphical User Interfaces, rather than discuss about the likelihood of a clone detection match. The only case law we found regards type 4 clones (indeed, with a negative outcome for the plaintiff). Apparently, courts tend to consider the broader picture of a program’s features. Even though there are, at the state of the art, excellent detection techniques⁵ Courts do not judge over heuristic techniques, even though they are considered highly reliable, thus preferring to judge over generic issues, like graphical interfaces.

This seems contradictory. If semantic clones are the most difficult ones to identify, why is all the available case law about these clones? Again, a reasonable explanation is that for a court it is easier to judge about the output of a code (like a GUI, or general functionalities) rather than deciding on which degree a code has been copied.

In the next three subsections we deepen our case cluster.

⁵For a comprehensive and recent survey, see [20].

The case law is represented in the table US Case Law. Please consider that we put together court's statements regarding cloning issues.

In Subsection 4.4 we analyzed cloning issues related to physical devices which may not be strictly related to software cloning. However, we found it interesting, because it shows how courts deal with cloning of technological issues. This gives us some insights about court's behavior. In Subsection 4.5 courts deal with the well known issue of antitrust and monopoly in the technology market. Here, judges weight the interests of free market competition and individual propriety rights. Subsection 4.6 is the more straightforward case of software cloning and copyright infringements.

4.4 Software and hardware cloning related to physical devices

Interestingly, cases belonging to the first cluster A (Software and Hardware cloning issues related to physical devices) are all concentrated before 2000, so among the first ones. Between 1990 and 2000 almost all cases were about software and hardware cloning of phone and television devices. Out of 18 relevant cloning cases in the last decade of XX century, 12 were related to such issues⁶ Only in 2012, in *United States v. Harris*, a US court treated a cloned device issue.

However, this case is related to a cloned software and hardware tool that enables a free and faster Internet access.

In *United States v. Davis* of 1992, Davis was convicted of violating various federal statutes and copyright infringement regarding cable television and its satellite-signal system. Once completed, Davis's modifications made it possible for the cloned modules to descramble and decrypt satellite programming without the knowledge of the cable companies. The modifications also made it all but impossible to use the device in any legitimate fashion.

Similarly, in *United States v. Yates* of 1995, the court held that cloning involved reprogramming a cellular telephone so that its electronic serial number and mobile identification number combination was identical to a legitimate customer's account. The court ruled that the defendant violated the law because cloning involved the use of an altered telecommunications instrument to obtain access to pay services for the purpose of defrauding the carrier.

In these cases, the outcome for the defendant was rather negative. Courts defined these cases within other legal domains (e.g., telecommunication law), with crimes clearly defined by the law.

4.5 Software cloning related to competition and antitrust issues

Case Law of the B cluster (Software cloning issues related to competition and antitrust issues) started at the very beginning of the XXI century. These cases are interesting because they show how courts (re)act to monopolistic behaviors. Usually, functional cloning is permitted (also encouraged) to create competition.

⁶For a comprehensive overview please cfr. the table US Case Law.

The most relevant cases involve Microsoft Corporation. Probably, the most interesting case is the first one: the 1999 *United States v. Microsoft Corp.* The US government claimed that Microsoft violated antitrust provisions of the Sherman Act, which is the most relevant antitrust law. The court concluded that Microsoft had monopoly power in the strategic market of PCs because the defendant could substantially raise its prices without losing business to a commercially viable alternative, since Microsoft's market share was large and stable, and the related market was protected by a high barrier of entry. The court further found that Microsoft purposefully leveraged its monopoly power in the market to thwart competition in other software markets. Specifically, through restrictive OS licensing agreements with computer manufacturers, Microsoft achieved a higher market share in the web browser market. Microsoft protected its monopoly and hindered innovation by imposing barriers to entry against various cross-platform software, "middleware", and network applications.

Here, cloning does not have a major importance, since the court intended to protect market competition. In fact, in cluster B cases, usually courts tend to see cloning as positive, since it creates competition, which is more important than property rights (e.g., copyright) [16]. Antitrust is one of the most studied subjects in Law and Economics literature [8]. Courts tend to restore equilibrium in the market, which is a common good.

Therefore, any possible act of competition is encouraged (directly or indirectly), even functional cloning.

Microsoft was sued by several other public bodies, for very similar accuses, like *New York v. Microsoft Corp.* in 2002 or *Massachusetts v. Microsoft Corp.* in 2004. Also common in cluster B are the cases of private corporations suing other corporations for unfair competition. In *Sun Microsystems, Inc. v. Microsoft Corp.* in 2000, the court granted a preliminary injunction, finding that Sun had a reasonable chance of success on the merits, the hardship to Sun of Microsoft's continuing its potentially unfair competition outweighed the burden on Microsoft, and Microsoft was likely to continue harming Sun if the injunction were not granted.

4.6 Software cloning related to misappropriation of trade secrets and copyright infringements

The last cluster - cluster C - of rulings related to misappropriation of trade secrets and copyright infringements is, probably, the most connected to software issues. It includes cases from 1990 to 2014.

We found no clear trend followed by the courts, because they protect the plaintiff or the defendant, case by case. More precisely, we cannot clearly state that courts tend to protect inventors from software cloning.

The oldest case in this cluster is *Lotus Development Corporation v. Borland International, Inc.* of 1990. The court dealt with generic issues, i.e., whether the copyright does extend to the text or layout of a program's menus, stating that it is not extendable. The court held that if the expression of an idea had elements that went beyond all

functional elements of the idea itself, and beyond the obvious, and if there were numerous other ways of expressing the non-copyrightable idea, then those elements of expression, if original and substantial, were copyrightable.

Also the *FASA Corp. v. Playmates Toys* case of 1994 is related to the cloning of playing symbols. The court indicated that a comparison of the game materials revealed that there were marked similarities between the two sets of playing symbols programmed in the games, but no substantial evidence of copyright infringement.

On the contrary, in *United States v. Manzer*, the jury determined that computer programs sold by Manzer were derivative of copyrighted material, and that the software contained sufficient notice of its protected status. Likewise, *Tradescape.com v. Shivaram* of 1999. Tradescape sued Shivaram for copyright infringement and theft of trade secrets concerning online day trading computer software. Shivaram, a software consultant that used to work for Tradescape, developed a software program that allowed for online day trading. Tradescape established a likelihood of success on the merits on its copyright infringement and theft of trade secrets claims because it provided sufficient direct and circumstantial evidence of copying of protected material.

Another important case is *Oracle Am., Inc. v. Google Inc.* of 2012. Oracle wrote 37 packages of Java source code, published their “application programming interfaces” (API), and licensed them to others for writing “apps” for computers, tablets, smartphones, and other devices. Oracle alleged that Google’s Android mobile operating system infringed Oracle’s patents and copyrights. The jury found no patent infringement, but ruled that Google infringed copyrights in the 37 Java packages and a specific routine, “rangeCheck”. Hence, copyright protection extends to all elements of an original work of computer software, including a system or method of operation, that an author could have written in more than one way.

The 14 cases in cluster C are interesting because there is not a clear positive or negative jurisprudential trend followed by the courts. Evaluating case by case, courts tend to value positively or negatively cloning issues. Usually, there are considered much more elements than just a simple “copied” or “non copied” answer. There is always a human, arbitrary element in any judgment. Thus, any kind of reasonable expectations of an outcome of similar cases are rather difficult.

4.7 Other cases related to the United States

In this paper we study the behavior of courts when they have to deal with software cloning. However, some cases of software cloning are not classified “software & cloning” inside legal databases. Therefore, analyzing all cases of software cloning, strictly speaking, is nearly impossible, since we should analyze any law case where software is involved. Moreover, even after the analysis of such cases, still you have not the guarantee that you took all, since not all cases are imported in such databases.

By serendipity search we found five other relevant cases where software cloning is involved. These cases are listed and explained in Table 8. They are all related to copyright infringements or other infringements. Since they do not fall

in the chosen query “software cloning”, we kept it separately. These five cases, *de facto* confirm our previous analysis.

The use of graphical user interface (GUI) elements that are similar between two systems were examined *Apple Computer, Inc. v. Microsoft Corporation* and *Hewlett-Packard Co.*; the court stated that such a (re)use does not represent an infringement. These claims do not deal with clone type 1, 2, or 3 but only with functional ones.

We believe that US courts do not clearly protect a copyright holder. In fact they have a waving aptitude, from case to case, according, mainly, to jurisprudential issues. This is not the case of the European Court of Justice case law, discussed in the next section.

Table 8: Other US case law.

Case	Court	Year	Relevant output
<i>Step-Saver Data Systems, Inc. v. Wyse Technology</i>	3rd Cir.	1991	Characterizing the transaction as a license to use software is a habit which do not correspond to today’s world.
<i>Computer Associates Int. Inc. v. Altai Inc.</i>	2d Cir.	1992	to claim for a copyright infringements there has to be claimed substantial similarities.
<i>Lewis Galoob Toys, Inc. v. Nintendo of America, Inc.</i>	9th Cir.	1992	Any consumer can do changes to licensed computer games for personal use.
<i>Apple Computer, Inc. v. Microsoft Corp.</i>	9th Cir.	1994	Certain components of a GUI do not underly to copyright law.
<i>Meshwerks, Inc. v. Toyota Motor Sales U.S.A., Inc., et al.</i>	10th Cir.	2008	3D models are not protected by copyright even if they represent the original.

4.8 The European Union case law

Only four cases relate to software cloning. The others deal with jurisprudential issues of abuse of dominant position⁷. Other cases regards patent or copyright registration issues. Since the Office for Harmonisation in the Internal Market (OHIM) is a European agency, any cases regarding such issues is of competence of the ECJ. Moreover, we have some cases regarding data protection and public competition against the EU issues.

Regarding the case law, the ECJ has relevant rulings regarding software cloning just for type 4. So, easily detectable clones (of type 1, 2, or 3) are not relevant at for the EU court.

What distinguishes the EU from the US is the different approach to copyright. The ECJ fully recognizes the legal

⁷Consider that the European Commission was established to create the Single Market. Actions against abuse of dominant position are among the most frequent ones, to foster the European competition [9].

dignity of computer programs as copyrightable goods (Used-Soft GmbH v. Oracle International Corp.). However, at the same time, it gave a loose protection to functional cloning (Bezpečnostní softwarová asociace v Ministerstvo kultury.). Moreover, the ECJ went way beyond with the SAS Institute Inc. v World Programming Ltd. case, which has several disruptive elements regarding copyright law, which will be analyzed in the next section.

Furthermore, consider that these sentences are a Supreme Court pronouncement, thus binding for all Member State jurisdictions.

Both US and EU courts never judged about type 1, 2, or 3 clones, which are relatively easy to detect with state of the art heuristic tools. The available case law concerns only type 4 clones (semantic ones) and other, general, copyright issues. Synthetically, we can say that:

- courts have an apparent difficulty to deal with clone detection issues;
- we did not find one single law case about clones of type 1, 2, or 3;
- the only cases treated by courts concern type 4 (functional) clones;
- the few cases regarding functional clones have, within the EU, a ‘loose’ copyright protection.

5. AN ECJ DISRUPTIVE RULING

After the SAS Institute v. World Programming Ltd⁸ ruling by the European Court of Justice the legal consideration of the EU justice system about copyright of software changed radically.

This ruling introduced diverse major consequences about how to interpret software, from a legal perspective. The implications for software programs and related lawsuits are of greatest relevance for the software community in Europe.

Going into details, the ECJ stated three important principles regarding the interpretation of Directives 91/250 and 2001/29:

1. The first, and most important part of this ruling is that the Court, stating that *the legal protection of computer programs is to be interpreted as meaning that the functionalities of a computer program and the programming language are not eligible, as such, for copyright protection. It will be for the national court to examine whether, in reproducing these functionalities in its computer program, the author of the program has reproduced a substantial part of the elements of the first program which are the expression of the author’s own intellectual creation.* So, since software’s features are considered as “principles” or “ideas,” by the Court they are not copyrightable expressions by themselves. The copyright of software is so, no more considered an absolute assumption but a relative one. This does not mean that computer programs are not copyrightable.

They are a form of expression of the intellectual creation of the programmer but the “principles” or “ideas,” themselves are not protected by law.

2. Moreover, regarding reverse engineering for interoperability issues, the ECJ affirms that *it is not regarded as an act subject to authorization for a licensee to reproduce a code or to translate the form of the code of a data file format so as to be able to write, in his own computer program, a source code which reads and writes that file format, provided that that act is absolutely indispensable for the purposes of obtaining the information necessary to achieve interoperability between the elements of different programs. That act must not have the effect of enabling the licensee to re-copy the code of the computer program in his own program, a question which will be for the national court to determine. [...] Acts of observing, studying or testing the functioning of a computer program which are performed in accordance with that provision must not have the effect of enabling the person having a right to use a copy of the program to access information which is protected by copyright, such as the source code or the object code.* The direct consequence of this statement is that any software engineer, who acquired a license of a software can freely observe, study or test it to fix interoperability or for education purposes. So, any software, which has been acquired legally, can be studied and the copyright holder is not able to prevent it. Interestingly, even though someone would study the program, to copy it, this could not be considered a copyright infringements.
3. The last paragraph of the ruling, which is less relevant from our perspective, is about the copyright of the user manual. According to the ECJ, *the reproduction, in a computer program or a user manual, of certain elements described in the manual for another computer program may constitute an infringement of the copyright in the latter manual if – a question which will be for the national court to determine – the elements reproduced in this way are the expression of their author’s own intellectual creation.* Also for this case, the expression, original creation of the author, is protected by copyright law. Not protected are *keywords, syntax, commands and combinations of commands, options, defaults and iterations* singularly, but *the choice, sequence and combination of such elements that the author may express his creativity in an original manner and achieve a result which is an intellectual creation.*

The disruptive nature of this ruling is quite clear. Even though before there were no real case law regarding cloning or copying issues, about copyright infringements, this sentence has a big impact in the computer science community because it states relevant issues that have a direct impact for programmers, at least in Europe. With the SAS judgment we can figure out the following direct consequences for software engineers:

- it is possible to reproduce “principles” or “ideas” of other people’s software programs;

⁸C-406/10, 02.05.2012

- it is possible to profit from others’ “principles” or “ideas,” since they are not protected by copyright;
- my own “principles” or “ideas” are not copyrightable, so everyone can get full inspiration from them;
- European courts are not the right place where to defend “principles” or “ideas” because no legal paradigm protects them;
- courts within the EU may intervene if the source or object code itself is copied;
- the source or object code of any program can be studied, without any permission of the licensor for “study” purposes. Therefore, if someone studies the source code of a program to get its “principles” or “ideas” to exploit them, no one can, *de facto* prevent it. Even though the court handles with the case which regard reverse engineering for interoperability issues, it is easy to bypass this case. In principle, any programmer could claim to have “studied” it for interoperability. So, even if nothing would come out (in terms of interoperability) still the programmer could have been studied the program, without any restriction. Finally, no one could claim copyright issues if he gets “principles” or “ideas” of that program, for his own program.
- it is legal (at least in Europe) to copy/clone “principles” or “ideas” of any program, also for profit.

6. IMPACT ON SOFTWARE

This papers shows how an interdisciplinary approach to software may bring enriching elements to the community discussion. In this case, bridging legal considerations within the clone literature brings some relevant insights about an everyday aspect of software: its legal protection. Here, we represented the behavior of US and EU courts when it comes to software cloning and, more generally, to IPR issues. In Table 9 we outlined the main differences between the two legal systems, with respect to software cloning.

For the US we found out that, at the moment, the Supreme Court has not issued a ruling regarding cloning issues in software. This means that first and second level courts will continue to use the pattern identified and discussed in Subsection 4.3. So, new cases of software cloning in the US will fall under cluster A, B or C.

For all EU Member States, the SAS Institute v. World Programming Ltd ruling will have a disruptive impact regarding software cloning issues. What we define as functional cloning or type 4 clones are perfectly admitted in all thinkable use. So, from academic, non profit or FOSS application, up to commercial, closed source and business applications, functional cloning has to be considered, at least permitted. The use of “principles” and “ideas” of software is free and may be used for the proper purposes. No one can legally claim an IPR protection if someone else exploit the own original “principles” or “ideas” used in someone else artifact.

What the community needs to understand very clearly is that all courts of EU’s Member States are not the suited places for a stringent protection of software artifacts. Such a conclusion has a wide overall impact on society.

Table 9: Copyright protection within the EU and the US

	EU	US
Difficulties to deal with cloning issues	✓	✓
Courts do not judge over cloning cases of type 1,2 or 3	✓	✓
Clear clustering of case law	X	✓
Software is patentable	X	✓
The use of “principles” or “ideas” belonging to other people is permitted	✓	X
It is possible to profit from others “principles” or “ideas”	✓	X
Courts are the right place to defend your own “principles” or “ideas”	X	✓

7. CONCLUSIONS

In this paper we analyzed the difference in behavior of US and EU courts.

Courts, usually, do not enter in cloning issues concerning program fragments that are identical. The case law we found is all is about semantic clones, which only clone detection heuristics can figure out. However, courts seem not to rely on such techniques, using, for their judgments, more general principles, like the use of software as method of operations.

The US courts have a waiving attitude and decide over software’s copyright protection case by case. Furthermore, no Supreme Court ruling relevant for software cloning issue was found.

The approach of the European Court of Justice appears looser in terms of degree of legal protection of software’s copyright. Undoubtedly, the SAS judgment has to be considered as disrupted in term of copyright protection. This has a wide relevance not only for the software engineering community but also for everyone dealing with intellectual propriety. According to the ECJ position, all Member States courts within the European Union, have to align their future rulings, since it is a pronouncement of a higher court. The Court stated that cloning of “principles” or “ideas” (semantic clones) can not be an infringement of copyright, since “principles” or “ideas” are not copyrightable.

Therefore, the question “how much is too much”, at least for courts, makes little sense. Both, US district and federal courts and the ECJ disregards cloning cases, strictly speaking i.e., type 1, 2 or 3. The only relevant cases are functional cloning ones.

A further interdisciplinary research effort may investigate future rulings of lower court within the EU. So, we could analyze the level and type of interpretation given by these courts to the SAS ruling. What we want to state is that the right interpretation of interdisciplinary issues in software engineering, like the effect of courts into the protection of the propriety of software is crucial, since it has a big impact on the community on both copyright and commercialization aspects. In particular, it could be of interest to analyze how European software houses complies with this new leaning

of courts. Moreover, a wider analysis of more legal systems (e.g., India, China) could give us better insights about court's behavior in cloning cases.

8. ACKNOWLEDGMENTS

The research presented in this paper has been funded partially by the ECSEL project MANTIS (662189).

9. REFERENCES

- [1] Baker, B.: On finding duplication and near-duplication in large software systems. Proceedings of the 2nd Working Conference on Reverse Engineering (WCRE 95), Toronto, Ontario, Canada, 86–95 (1995)
- [2] Bergstra, J., Klint P.: About "trivial" software patents: The IsNot case. *Science of Computer Programming*. 64, 264–285 (2006)
- [3] Ciancarini, P., Favini G.P.: Detecting clones in game-playing software. *Entertainment Computing*. 1, 9–15 (2009)
- [4] Ciancarini, P., Russo D., Sillitti, A., Succi, G.: Reverse Engineering: a European IPR perspective. 31st ACM Symposium On Applied Computing (SAC 2016). Pisa, Italy (2016).
- [5] Di Penta, M, German, D. M., Guéhéneuc, Y., Antoniol, G.: An exploratory study of the evolution of software licensing. Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering, 145–154, (2010)
- [6] German, D. M., Hassan, A. E.: License integration patterns: Addressing license mismatches in component-based development. Proceedings of the 31st International Conference on Software Engineering, 188–198 (2009)
- [7] Gross, H.G., Melideo, M., Sillitti, A.: Self Certification and Trust in Component Procurement. *Journal of Science of Computer Programming*, Elsevier, 56, 141 – 156 (2005).
- [8] Hylton, K., N.: *Antitrust Law and Economics*. Edward Elgar Publishing Lim. (2010)
- [9] Horspool, M., Humphreys M.: *European Union Law*. Oxford University Press (2012)
- [10] Johnson, J.H.: Substring matching for clone detection and change tracking. Proceedings of the 10th International Conference on Software Maintenance, Victoria, British Columbia, Canada, 120–126 (1994)
- [11] Kapser, C.J., Godfrey M.W.: "Cloning considered harmful" considered harmful: patterns of cloning in software. *Empirical Software Engineering*, 13, (6), 645–692, (2008)
- [12] Kapser, C.J., Godfrey M.W.: Supporting the analysis of clones in software systems: a case study. *Journal of Software Maintenance and Evolution: Research and Practice*. 18, (2), 61–82 (2006)
- [13] Kim, M., Bergman, L., Lau, T., Notkin, D.: An Ethnographic study of copy and paste programming practices in OOPL. Proceedings of 3rd International ACM-IEEE Symposium on Empirical Software Engineering (ISESE 04), Redondo Beach, CA, USA, 83–92 (2004)
- [14] Koschke, R.: Frontiers of software clone management. Proceedings of Frontiers of Software Maintenance (FoSM'08), Beijing, China, 119–128 (2008)
- [15] Lavoie, T., Eilers-Smith, M., Merlo, E.: Challenging cloning related problems with GPU-based algorithms. Proceedings of 4th International Workshop on Software Clones, Cape Town, SA, 25–32 (2010)
- [16] Mc Nutt, P., A.: *Law, Economics and Antitrust: Towards a New Perspective*. Edward Elgar Publishing Lim. (2005)
- [17] Monden, A., Nakae, D., Kamiya, T., Sato, S., Matsumoto, K: Software quality analysis by code clones in industrial legacy software. Proceedings of 8th IEEE International Symposium on Software Metrics (METRICS 02), Ottawa, Canada, 87–94 (2002)
- [18] Petrinja, E., Sillitti, A., Succi, G.: Comparing OpenBRR, QSOS, and OMM Assessment Models. 6th International Conference on Open Source Systems (OSS 2010), Notre Dame, IN, USA (2010)
- [19] Predonzani P., Sillitti, A., Vernazza, T.: Components and Data-Flow Applied to the Integration of Web Services. The 27th Annual Conference of the IEEE Industrial Electronics Society (IECON'01), Denver, CO, USA, 29 November - 2 December (2001)
- [20] Rattan, D., Bathia, R. Singh, M.: Software clone detection: A systematic review. *Information and Software Technology*. 55, 1165–1199 (2013)
- [21] Remencius, T., Sillitti, A., Succi, G.: Assessment of software developed by a third-party: A case study and comparison: Information Sciences, Elsevier, Vol. 328, 237 – 249 (2016)
- [22] Rieger, M., Ducasse, S., Lanza, M.: Insights into System - Wide Code Duplication. WCRE, Delft University of Technology. 100–109 (2004)
- [23] Riis, S.: What makes a chess program original? Revisiting the Rybka case. *Entertainment Computing*. 5:3, 189–204 (2014)
- [24] Roy, C.K., Cordy, J.R.: A Survey on Software Clone Detection Research. Queen's School of Computing Tech. Report 2007–541, Kingston, 115 (2007)
- [25] Runeson, P., Höst, M.: Guidelines for conducting and reporting case study research in software engineering *Empirical software engineering*. 14 (2), 131–164 (2009)
- [26] Succi, G., Succi G, Ronchetti M.: Legal Issue Regarding Software Use and Reuse within the European Union Legislation. *Journal of Computing and Information Technology*. 4:3, 179–186 (1996)
- [27] van den Herik H.J., Plaat A., Levy D., Dimov D.: Plagiarism in game programming competitions. *Entertainment Computing*. 5:3, 173–187 (2014)