13. Licencjonowanie oprogramowania

Tomasz Boiński i Szymon Olewniczak

Katedra Architektury Systemów Komputerowych
Wydział Elektroniki, Telekomunikacji i Informatyki
Politechnika Gdańska, Gdańsk
tomasz.boinski@eti.pg.edu.pl
szymon.olewniczak@pg.edu.pl

Streszczenie
Wolne i otwarte oprogramowanie przeżywa ostatnimi laty rozkwit. Coraz więcej przedsiębiorstw komercyjnych opiera rozwój swoich firm na otwartym oprogramowaniu. Zarówno mali, jak i duzi gracze mają świadomość komplikacji współczesnych systemów i niemożności samodzielnego ich rozwoju. Z pomocą przychodzi otwarte podejście do wytwarzania oprogramowania. Wymaga to jednak pewnego zrozumienia uwarunkowań prawnych, a w szczególności licencji, na jakich wydawane jest oprogramowanie. Niniejsze opracowanie ma na celu omówienie tychże uwarunkowań, odpowiada na pytanie dlaczego potrzebujemy licencji na oprogramowanie. Opisywane zostały również najważniejsze wg autorów licencje, oraz wskazuje, czym należy kierować się przy doborze licencji.

Słowa kluczowe: oprogramowanie, licencje, prawo autorskie.

13.1 Wstęp

Ostatnimi laty zaobserwować można rosnące zainteresowanie wykorzystaniem otwartych komponentów programowych. Wolne i otwarte oprogramowanie wspierane i wykorzystywane jest przez praktycznie wszystkich większych graczy na rynku, nawet tych, którzy do tej pory zaciekle zwalczali ruch Free/Open Source (jak np. Microsoft).

Rosnąca akceptacja dla wolnego i otwartego oprogramowania wynika z kilku faktów. Przede wszystkim z komplikacji współczesnych systemów czy aplikacji, a tym samym rosnących kosztów jego wytwarzania. Drugim istotnym aspektem jest postępująca migracja modeli biznesowych ze sprzedaży oprogramowania na sprzedaż usług. W tym pierwszym wypadku zastosowanie licencji wolnych oraz otwartych miało poważny wpływ na licencje finalnego produktu. Warunki licencyjne narzucane przez otwarte komponenty mogły być nie do zaakceptowania dla wielu firm. W przypadku sprzedaży usług samo oprogramowanie jest tylko elementem modelu biznesowego, często bezwartościowego bez zasobów kadrowych, sprzętowych czy know–how przedsiębiorstwa.

W niniejszym rozdziale omówione zostały uwarunkowania prawne, odpowiadające na pytanie, dlaczego potrzebujemy licencji na oprogramowanie (część 13.2).

Następnie w części 13.3 przedstawiono podstawowe różnice pomiędzy licencjami wolnymi i otwartymi. Część 13.4 opisuje najważniejsze wg autorów licencje na oprogramowanie. W części 13.5 zaprezentowano zagadnienie zgodności licencji, a następnie część 13.6 wskazuje, czym należy kierować się przy doborze licencji. Całość wieńczy krótkie podsumowanie.

13.2 Po co nam licencje?

Oprogramowanie, jak każdy utwór, z definicji chroniony jest prawem autorskim, które definiuje ramy wykorzystania oprogramowania, w jakich poruszać może się użytkownik końcowy. Przykładowo art. 75 ust. 1. ustawy o prawie autorskim i prawach pokrewnych [12], jeżeli umowa nie stanowi inaczej, zezwala przy pewnych warunkach (np. gdy kopia nie będzie używana równolegle z oryginałem) użytkownikowi końcowemu na sporządzenie kopii zapasowej oprogramowania, na analizę konstrukcji i zasady działania oprogramowania, zwielokrotnianie kodu lub tłumaczenie jego formy, jeżeli jest to konieczne dla umożliwienia współpracy z oprogramowaniem zewnętrznym. Analogicznie np. w USA, sekcja 117 Copyright Act zezwala właścicielowi kopii oprogramowania na wykorzystywanie tej kopii za pomocą komputera, nawet jeżeli to wykorzystywanie będzie wymagało wykonania kopii lub modyfikacji oprogramowania. Kopiowanie takie czy też modyfikacja, nie wymaga więc zgody właściciela praw autorskich i zezwala na wykorzystanie kopii posiadanej kopii oprogramowania bez licencji. W ogólności więc właściciel kopii oprogramowania może wykonać kopię utworu na użytek własny, a nawet podzielić się nią z osobami pozostającymi z nim w kręgu osobistym.

Zapisy prawa autorskiego stoją w sprzeczności zarówno z interesem komercyjnych dostawców oprogramowania (możliwość analizy, badania i modyfikacji kodu, prawo do wykonywania kopii), jak i przedstawicieli ruchu wolnego i otwartego oprogramowania (np. brak możliwości tworzenia dzieł pochodnych czy redystrybucji zmian). Wprowadzono więc pojęcie licencji na oprogramowanie, oryginalnie w odpowiedzi na potrzeby komercyjnych dostawców oprogramowania.

Licencje komercyjne, będące de facto umową między dystrybutorem a nabywcą, posiadają więc zapis o utrzymanie własności kopii przez dystrybutora (stąd ich nazwa licencje własnościowe). Jako że odbiorca oprogramowania nie jest w ich rozumieniu właścicielem kopii, odpowiednie sekcje (np. art. 75 ust. 1 ustawy o prawie autorskim i prawach pokrewnych czy sekcja 117 Copyright Act) przestają obowiązywać, a użytkownik końcowy, chcąc korzystać z oprogramowania, musi zaakceptować ostrzejsze wymogi licencyjne, niżby to wynikało z prawa autorskiego. Otrzymywane oprogramowanie nie staje się jego własnością, a jedynie narzędziem niezbędnym do realizacji umowy.

Z kolei licencje wolne i otwarte w ogólności przekazują własność kopii, jednak ich akceptacja daje użytkownikowi dodatkowe prawa, np. prawo do modyfikacji oprogramowania i redystrybucji tychże zmian. Należy tutaj jednak pamiętać, że właściciel kopii nie jest tożsamy z właścicielem praw autorskich. Użytkownik końcowy uzyskuje prawa, wynikające z ustawy, tylko do tej jednej kopii, reszta praw pozostaje przy właścicielu praw autorskich.

Licencje na oprogramowanie pełnią również dodatkowe funkcje. Oprócz nadawania praw czy wymuszania ograniczeń licencje opisują zależności oraz odpowiedzialność za kod i wykorzystanie oprogramowania zachodzące pomiędzy stronami porozumienia licencyjnego. Zazwyczaj mówią, że producent nie ponosi odpowiedzialności za wszelkie uszkodzenia wynikłe z prawidłowego jak i nieprawidłowego wykorzystania oprogramowania. Ograniczają się jedynie do stwierdzenia, że dystrybutor dołożył wszelkich starań do sprawdzenia poprawności oprogramowania, a korzystanie z niego zgodnie z jego zastosowaniem nie powinno spowodować start czy uszkodzenia sprzętu. W rozwiązaniach komercyjnych zazwyczaj dodatkowo chronią przedsiębiorstwo dystrybutora opisując zakres odpowiedzialności, np. za stan usług związanych z oprogramowanie.

13.3 Wolne a otwarte licencje

Wolne oraz otwarte licencje posiadają pewne cechy wspólne. W ogólności, w obu rodzajach licencji, prawo własności kopii przekazywane jest użytkownikowi końcowemu. Niesie to za sobą pewne istotne następstwo – akceptacja licencji jest opcjonalna – użytkownik końcowy może używać oprogramowanie bez konieczności akceptacji licencji, a tym samym może badać oprogramowanie czy np. wykonywać zmiany, ale jedynie na własne potrzeby. Jeżeli użytkownik chce skorzystać z dodatkowych praw wynikających z licencji (np. możliwość redystrybucji samego oprogramowania czy też zmian) to musi ją zaakceptować i przestrzegać.

Z punktu widzenia użytkownika końcowego licencje wolne i otwarte mogą nie być rozróżnialne. Warunki typowe dla każdego z rodzajów licencji niosą za sobą dość istotne następstwa w kwestii możliwości wykorzystania oprogramowania wydanego na poszczególnych licencjach.

Licencje wolne mają na celu zachowanie wolności oprogramowania. Wolność użytkownika jest pochodną tejże. Licencje te dają końcowemu użytkownikowi duże uprawnienia, takie jak możliwość redystrybucji, inżynierii wstecznej czy innej formy modyfikacji kodu. Zazwyczaj dodatkowe prawa wymagają czegoś w zamian od użytkownika, musi się on zgodzić na dodatkowe warunki, by móc z tych praw skorzystać, np. licencje tego rodzaju zazwyczaj wymuszają, by wszelkie modyfikacje czy dzieła pochodne dystrybuowane były na tej samej licencji co oryginalne dzieło. Warunek ten często ogranicza możliwe pola zastosowania oprogramowania (np. prawnie nie możemy takiego oprogramowania włączyć do aplikacji komercyjnej, a następnie redystrybuować na licencji własnościowej). Tym samym również uniemożliwiają łączenie oprogramowania z niektórymi innymi otwartymi licencjami. Najbardziej znanym przedstawicielem licencji wolnych jest licencja GPL [1].

Licencje otwarte dzielą się na dwa rodzaje:

1. 'copyleft' – analogicznie jak licencje wolne mają na celu zachowanie wolności oprogramowania. Cechują się podobnymi atrybutami, w ogólności każda wolna licencja to licencja otwarta typu 'copyleft'. Najbardziej znanym typem tej licencji jest licencja GPL [1];

2. licencje „zezwalające" (ang. *permissive licenses*) – mają na celu zachowanie wolności końcowego użytkownika. Zezwalają końcowemu użytkownikowi na pełną dowolność w wykorzystaniu oprogramowania nie nakładając związanych z tym obowiązków. Oprogramowanie na takiej licencji może być bez przeszkód wykorzystywane przez oprogramowanie własnościowe. Przykładem takiej licencji są np. licencje BSD [14, 15] czy MIT [16].

13.4 Najistotniejsze wolne i otwarte licencje

Istnieje wiele licencji zaliczanych zarówno do grupy licencji wolnych, jak i otwartych [28]. Część z nich stosowana jest powszechnie, część jedynie w dedykowanych projektach (np. licencja PostgreSQL [25] czy też Mozilla Public License [24]). Należy również pamiętać o domenie publicznej (ang. *public domain*). Nie jest to licencja, jednak w praktyce działa tak, jakby to była jedna z nich. Oprogramowanie znajdujące się w przestrzeni publicznej może być dowolnie używane, jest jednak rzadkością – do domeny publicznej zazwyczaj trafiają dzieła określony czas (70–100 lat) po śmierci oryginalnego autora. W przypadku oprogramowania musi być ono więc jawnie umieszczone w przestrzeni publicznej (taką formę licencjonowania przyjęło np. SQLite [26]). Jednakże na przestrzeni najbliższych 50–100 lat lawinowo oprogramowanie zacznie trafiać do domeny publicznej. W dalszej części niniejszego rozdziału zaprezentowane zostaną najpopularniejsze licencje, powszechnie wykorzystywane przez społeczność.

13.4.1 MIT/X11

Jest to prosta, zezwalająca licencja, zgodna z GPL [16]. Oryginalnie stosowana była głównie przez projekt XFree86 (najpopularniejsza implementacja systemu X Window dla systemów UNIX'owych). Licencja ta hołduje ogólnej zasadzie: „Możesz rozbić co tylko zechcesz z kodem, ale nie możesz powiedzieć, że go napisałeś". W rezultacie oprogramowanie wydane na licencji MIT/X11 może być wykorzystywane w zamkniętym kodzie, pod warunkiem dołączenia treści licencji.

13.4.2 BSD

Kolejna prosta, powszechnie wykorzystywana zezwalająca licencja. Istnieją jej cztery główne wersje:

- Oryginalna, czterozdaniowa licencja zezwalająca na dowolne wykorzystanie kodu, ale zabraniająca używania w materiałach reklamowych nazwisk/nazw autorów i wymagająca wspomnienia o nich w dokumentacji [21]. Wersja ta nie jest zgodna z licencją GPL w wersji 2, ale jest zgodna z GPL w wersji 3.
- Zmodyfikowana, trzyzdaniowa wersja, zabraniająca używania w materiałach reklamowych nazwisk/nazw autorów [15]. Dla odróżnienia od oryginalnej licencji BSD występuje pod nazwami: „BSD License 2.0", „Revised BSD License", „New BSD License", lub „Modified BSD License". Jest zgodna z każdą wersją licencji GPL.

– Uproszczona, dwuzdaniowa wersja [14], znana pod nazwami: „Simplified BSD License" lub „FreeBSD License". Nie posiadająca ograniczeń narzucanych przez pierwsze 2 wersje. Jest zgodna z każdą wersją licencji GPL.
– zerozdaniowa [17], zwana również „BSD Zero Clause License". Podobnie jak wersja 2-zdaniowa nie nakłada żadnych ograniczeń, dodatkowo nie wymaga umieszczania jakiejkolwiek informacji w kodzie, dokumentacji czy wersji binarnej dzieła pochodnego. Jest zgodna z każdą wersją licencji GPL.

Jest to powszechnie wykorzystywana rodzina licencji, jednakże nie zaleca się używania jej oryginalnej, czterozdaniowej wersji. W tym przypadku, jako że wszystkie wersje funkcjonują pod jedną wspólną nazwą, należy każdorazowo zwrócić uwagę na dokładne zapisy licencyjne. Stąd też Free Software Foundation zaleca, by zamiast licencji BSD używać jednoznacznej co do nazewnictwa licencji MIT/X11 [10].

13.4.3 Apache 2.0

Licencja [22] ta wymaga zachowania informacji o posiadaczu praw autorskich, zezwala na stosowanie oprogramowania w trakcie wytwarzania zarówno zamkniętego, jak i wolnego czy otwartego kodu. Nie jest więc licencją typu 'copyleft'. Jest to jedna z niewielu licencji dotykająca również problemu patentów. W jej myśl każdy współtwórca udziela automatycznie licencje na patenty implementowane przez to oprogramowanie. Dzięki temu licencja jest dość powszechnie wykorzystywana przez podmioty komercyjne. Warunek też powoduje, że licencja ta jest zgodna z GPLv3 (ale nie vice versa), ale nie jest zgodna z GPLv1 and GPLv2.

13.4.4 GPL

Chyba najbardziej znana wolna i otwarta licencja [1]. Została opracowana przez Richarda Stallmana, jest emanacją filozofii stojącej za Free Software Foundation. Użytkownikowi końcowemu, który zaakceptował licencję, daje ona szerokie możliwości wykorzystania kodu. Użytkownik ma więc prawo do dowolnego wykorzystania oprogramowania, badania zasady jego działania, modyfikacji oraz dalszej redystrybucji zarówno oryginału, jak i wprowadzonych zmian. Uprawnienia te nie są jednak bezwarunkowe, jak to miało miejsce we wcześniej omawianych licencjach. Jako że jest to licencja typu 'copyleft', wymaga by każde dzieło pochodne dystrybuowane również było na dokładniej tej samej licencji. Każdy odbiorca oprogramowania (w dowolnej formie, również binarnej) musi więc otrzymać pełną kopię kodu źródłowego.

Dzieło pochodne jest tu rozumiane bardzo szeroko – każde oprogramowanie, które wykorzystuje, łączy, zmienia, czy nawet linkuje (zarówno dynamicznie, jak i statycznie) kod wydany na licencji GPL, jeżeli tylko jest rozpowszechniane, musi być wydane na tej samej licencji [9]. Ma to na celu zachowanie przechodniości i niezmienności praw wynikających z tejże licencji w sytuacji, gdy dzieło zostało zmienione lub rozbudowane. W praktyce przyjmuje się więc (nawet mimo pewnych dywagacji prawnych dotyczących np. europejskiego systemu

prawnego [20]), że jeżeli tylko oprogramowanie jest dystrybuowane i wykonywane w tym samym procesie co oprogramowanie wydane na licencji GPL, niezależnie od związku między poszczególnymi fragmentami tego oprogramowania, całość musi być wydana na licencji GPL. Ma to szczególne znaczenie w sytuacji, gdy korzystamy z bibliotek systemowych, czy tworzymy np. rozszerzenia (tzw. *plug–in*). Ponadto obecnie funkcjonują 2 wersje licencji – 2 i 3. Nie są one ze sobą zgodne, a dzieło wydane na licencji np. w wersji 2 nie może być bez zgody autorów wydane na licencji w wersji 3 (patrz część 13.5)! Wynika to wprost z warunku, że każde dzieło pochodne musi być wydane na dokładnie tej samej licencji.

"Wirusowość" licencji GPL może rodzić pewne problemy z praktycznym zastosowaniem wydanego na niej oprogramowania. Wprowadzono więc szereg wyjątków od tej reguły. Dla przykładu implementacja openJDK środowiska Java posiada tzw. *CLASSPATH exception*. Problem polega na tym, że każda aplikacja napisana w języku Java jest kompilowana i wykonywana w obrębie procesu maszyny wirtualnej Java. Każdy kod jest również dziełem pochodnym po maszynie wirtualnej (chociażby poprzez dziedziczenie po klasie java.lang.Object). Wyjątek mówi więc, że jeżeli kod odwołuje się do bibliotek systemowych, zdefiniowanych w ramach maszyny wirtualnej, poprzez ich ładowanie za pomocą zmiennej środowiskowej CLASSPATH, to kod ten nie musi być wydany na licencji GPL. Z podobnych przyczyn zdefiniowano inne wyjątki, takie jak *libstdc++* (w celu umożliwienia kompilacji oprogramowania kompilatorem GCC i włączenie nagłówków bibliotek systemowych) czy *system* (w celu umożliwienia wykonywania wywołań systemowych).

Należy oczywiście pamiętać, że warunki licencji GPL obowiązują nas w przypadku, gdy udostępniamy komuś kopię oprogramowania. Sam fakt wykorzystania czy modyfikacji kodu wydanego na licencji GPL nie wywołuje obowiązku dystrybucji kodu, o ile nie nastąpiła dystrybucja oprogramowania w innej formie!

13.4.5 LGPL

GNU Lesser General Public License (LGPL) [8] jest to zmodyfikowana, bardziej zezwalająca wersja licencji GPL, oryginalnie pomyślana jako licencja dla bibliotek. Obecnie Free Software Foundation nie zaleca tego kroku ze względu na dysproporcje w liczbie bibliotek dostępnych na licencjach GPL i LGPL [6].

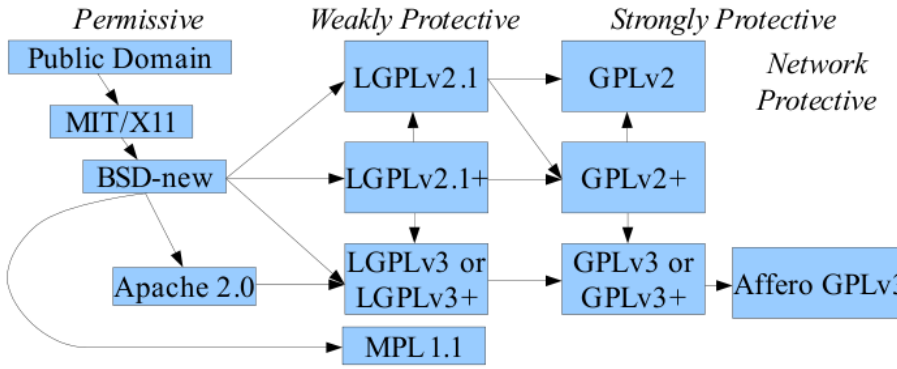
W odniesieniu do samego kodu licencja LGPL zachowuje się analogicznie jak licencja GPL – każdy odbiorca oprogramowania musi więc otrzymać kod źródłowy zarówno samego programu, jak i wprowadzonych do niego modyfikacji. Licencja LGPL pozwala jednak na łączenie kodu opartego na LGPL z kodem wydanym na dowolnej innej licencji, w tym komercyjnej. Warunkiem jest jednak, by kod LGPL dystrybuowany był niezależnie (np. jako biblioteka w systemie), a licencja oprogramowania zezwalała na modyfikację wersji wykorzystywanej biblioteki oraz reverse engineering potrzebny do debugowania tych zmian. W przeciwieństwie do GPL, linkowanie dynamiczne do biblioteki wydanej na licencji LGPL nie wymusza wykorzystania tejże licencji dla dzieła pochodnego.

13.4.6 AGPL

Affero General Public License (GNU AGPL) [7] jest tożsama z licencją GPL. O tyle o ile jednak licencja GPL obejmuje przypadki, kiedy użytkownik otrzymuje kopię oprogramowania, to licencja AGPL obejmuje również sytuacje, gdy użytkownik korzysta z oprogramowania jako z usługi bez otrzymania jego kopii (np. przez zdalny dostęp za pomocą przeglądarki internetowej). Licencja ta wymaga, by kod źródłowy aplikacji był dostępny dla każdego użytkownika, już na etapie logowania do aplikacji.

13.5 Zgodność licencji

Warunki nałożone przez licencje, zwłaszcza warunki nakładane na licencje dzieł pochodnych przez licencje wolne, w niektórych przypadkach mogą ograniczać czy wręcz uniemożliwiać mieszanie oprogramowania. Mówimy tutaj o tzw. kompatybilności licencji. Graf z rys. 13.1 obrazuje kompatybilność najpopularniejszych licencji. Strzałka od A do B oznacza, że można łączyć oprogramowanie wydane na licencji A z oprogramowaniem wydanym na licencji B, ale całość musi być wydana na licencji B.



Rysunek 13.1. Kompatybilność najpopularniejszych wolnych i otwartych licencji [27].

Licencje możemy tutaj podzielić na licencje zezwalające (ang. *permissive*), które nie nakładają praktycznie żadnych wymagań i mogą być łączone z dowolnymi innymi licencjami, nawet komercyjnymi. Kolejną warstwę stanowią tzw. licencje typu *weakly protective*. Licencje te dość silnie chronią bezpośrednio oprogramowanie na nich wydane, jednak pozwalają zazwyczaj na włączanie go do oprogramowania wydanego na innych licencjach. Kolejną warstwą są licencje typu *strongly protective*. Licencje te zazwyczaj nie umożliwiają łączenia oprogramowania z innymi licencjami, zazwyczaj z powodu konieczności udostępniania dzieła pochodnego na takiej samej licencji. Ostatnią, najsilniejszą warstwą są

licencje z grupy tzw. *network protective*. W tym przypadku działanie licencji jest bardzo szerokie, obejmuje również zdalny dostęp do oprogramowania, nie tylko jako bezpośrednio do kopii, a również jako usługi.

Zgodność licencji niesie za sobą istotne następstwa prawne rozstrzygające o możliwości (bądź nie) integracji oprogramowania. Licencje zezwalające w ogólności można włączyć do dowolnego innego oprogramowania wydanego na dowolnej licencji. Im licencja jest jednak bardziej „chroniąca" (ang. *protective*), tym mniejszy jest zakres dopuszczalnych do integracji licencji.

Szczególną uwagę należy zwrócić na licencje z rodziny licencji GPL. Każda z tych licencji wymaga, by dzieło pochodne wydane zostało na dokładnie takiej samej licencji, nawet z dokładnością do jej wersji. W praktyce oznacza to, że oprogramowanie wydane na licencji GPL w wersji 2 może być łączone tylko z oprogramowaniem wydanym na licencji GPL w wersji 2 lub innej, bardziej zezwalającej licencji. Nie ma możliwości, by oprogramowanie takie połączyć z wydanym na licencji GPL w wersji 3. Licencja ta wymaga wydania dzieła pochodnego również na licencji GPL w wersji 3 co stoi w sprzecznością wymogowi wydania tegoż na licencji GPL w wersji 2. Aby uniknąć tego problemu Free Software Foundation [2] zaleca, by używać dopisku „or later" (*lub późniejsze*, na rys. 13.1 oznaczone jako „+" przy nazwie licencji). Oznacza to, że autor wydając oprogramowanie na licencji np. „GPL wersja 2 lub późniejsze" w ogólności wydaje oprogramowanie na tejże wprost wskazanej licencji (GPL w wersji 2), jednak zgadza się również na włączanie oprogramowania do wydanego na dowolnej późniejszej wersji licencji GPL. Podejście to ma jednak pewną, zasadniczą wadę – zgadzamy się na wydanie naszego oprogramowania na licencji, której warunków nie znamy. Problem ten dało się zauważyć w chwili wprowadzenia wersji 3 licencji GPL, co do której część developerów wyrażało dość krytyczne stanowisko [4, 5, 19].

13.6 Wybór licencji

Wybór właściwej licencji ma duży wpływ na przyszłe zastosowanie wytworzonego przez nas kodu. Należy tutaj wziąć pod uwagę takie czynniki, jak nasz osobisty stosunek do upublicznianego kodu, zakres praw jakie chcemy udzielić, czy chcemy coś w zamian. W ogólności ważne jest również, by w ogóle jakąś licencję wybrać. W przeciwnym razie zakres możliwych zastosowań naszego kodu jest znacznie ograniczony. Nie możemy dla przykładu dystrybuować dzieł pochodnych.

Istotne jest również, by wziąć pod uwagę zgodność licencji oraz ich rozpoznawalność w środowisku. Część projektów tworzy własne licencje, które de facto są powtórzeniem już istniejących (np. licencja PostgreSQL [25]), co wzbudza zbędną dyskusję na temat jej natury, zasadności, konsekwencji czy zgodności z innymi licencjami [3, 11, 13, 18, 23]. Nierozważna zmiana treści licencji może naruszyć zasadę zgodności i doprowadzić wręcz do upadku projektu. W 2004, wraz z wydaniem XFree86 4.4, licencja MIT/X11 została zmodyfikowana, wzbudzając tym samym wielkie niezadowolenie w społeczności. Modyfikacja licencji polegała na dodaniu jednego, prostego warunku: „The end–user documentation

included with the redistribution, if any, must include the following acknowledgment: "This product includes software developed by The XFree86 Project, Inc (http://www.xfree86.org/) and its contributors", in the same place and form as other third–party acknowledgments. Alternately, this acknowledgment may appear in the software itself, in the same form and location as other such third–party acknowledgments.". W skrócie autorzy zażądali większego uznania, czy to w dokumentacji czy też w samym oprogramowaniu w postaci umieszczenia tamże stosownego zapisu. Tak zmodyfikowana licencja okazała się zgodna z GPL w wersji 3, ale nie z obowiązującą wówczas wersją 2. Narzucała również daleko idącą komplikację w utrzymaniu dokumentacji oprogramowania. Zmiana ta spowodowała zaprzestanie wykorzystania XFree86 przez wszystkie główne dystrybucje systemu Linux. Te, które tego nie zrobiły od razu (np. Mandrake), wycofały się z tej decyzji w przeciągu kliku miesięcy od wydania pierwszej wersji dystrybucji zawierającej kod na nowej wersji licencji.

Więc jaką licencję należy wybrać? Na to pytanie nie ma jednej słusznej odpowiedzi. Należy zastanowić się w dużej mierze, jakie mamy oczekiwania co do dalszego wykorzystania kodu oraz ewentualnych zobowiązaniach osób z niego korzystających.

Jeżeli napisaliśmy jakieś oprogramowanie i po prostu chcemy się nim podzielić z innymi, nie mamy nic przeciwko komercyjnemu wykorzystaniu go oraz nie zależy nam na ewentualnych poprawkach do tego kodu, najlepszą licencją będzie jedna z licencji zezwalających, takie jak MIT czy też dwuzdaniowa licencja BSD. Podejście to jest dobre również w sytuacji, gdy sami nie rozwijamy aktywnie kodu. FSF zaleca licencję MIT/X11 jako jednoznaczną (tożsama z nią licencja BSD ma co najmniej cztery wersje). Z drugiej jednak strony licencje BSD są bardziej precyzyjne. Licencja X11 zawiera nieprecyzyjne sformułowania „zajmować się oprogramowaniem" (ang. *to deal in the Software*) oraz „oprogramowanie i związane [dołączone] pliki dokumentacji" (ang. *this software and associated documentation files*), które mogą sprawiać problemy w interpretacji prawnej.

W sytuacji, gdy chcielibyśmy jednak utrudnić komercyjne zastosowanie naszego rozwiązania czy też chcemy mieć możliwość skorzystania z poprawek wprowadzonych przez użytkowników naszego kodu, należy zastosować licencję typu „copyleft", w szczególności właściwy wariant licencji GPL. W przypadku zgody na zastosowanie w rozwiązaniach komercyjnych warto również sięgnąć po licencję LGPL. W większości zastosowań wersja 3 licencji będzie odpowiednia. Wprowadza ona jednak pewne obostrzenia natury moralnej czy filozoficznej, np. brak możliwości wykorzystania oprogramowania wydanego na licencji GPL do implementacji mechanizmu DRM (ang. *Digital Rights Management*). Należy tutaj rozważyć, czy chcemy nakładać takie ograniczenia na użytkownika naszego oprogramowania.

W każdym przypadku musimy również pamiętać o patentach. Znaczna większość licencji w ogóle nie porusza tego problemu, wolne i otwarte oprogramowanie może więc być obciążone koniecznością posiadania licencji na patent, który został w tymże oprogramowaniu zaimplementowany. W sytuacji, gdy nasze oprogramowanie implementuje patent, do którego mamy prawa, warto zastosować licencję

Apache 2.0 lub (L)GPL w wersji 3. Licencje te jako jedne z niewielu poruszają kwestie patentów. Należy jednak pamiętać, że licencje z rodziny GNU są pod tym względem dość agresywne.

Jak już zdecydujemy się na właściwą z naszego punktu widzenia licencję, należy pamiętać, by właściwie oznaczyć projekt. Każda z licencji ma inne wymagania co do formy, w jakiej należy przekazać jej treść. W ogólności jednak użytkownik końcowy powinien otrzymać wraz z oprogramowaniem treść licencji. Ta najczęściej zawarta jest w pliku tekstowym, zwyczajowo nazwanym COPYING, umieszczonym w katalogu głównym projektu. Część licencji (przykładowo wszelkie odmiany licencji BSD) wymaga, by pliki z kodem źródłowym również oznaczyć stosowną informacją. Zazwyczaj czyni się to poprzez dodanie na początku każdego z plików komentarza zawierającego nazwę licencji oraz właściciela praw autorskich majątkowych. Często, zwłaszcza dla licencji krótkich, takich jak BSD czy MIT, przytacza się pełną treść licencji. Należy również zwrócić uwagę, że część licencji wymaga stosownego komentarza w wersji binarnej aplikacji. Niestety każdorazowo należy posilić się treścią samej licencji, gdyż nie ma tutaj ujednoliconej formy przekazywania stosownej informacji o licencji.

13.7 Podsumowanie

Wytwarzając oprogramowanie, oprócz kwestii jakościowych, musimy baczną uwagę zwrócić również na kwestie prawne. W sytuacji, gdy planujemy upubliczniać nasz kod, wybór właściwej licencji może być kluczowy dla potencjalnych użytkowników naszego rozwiązania, a czasami może wręcz uniemożliwić jego zastosowanie. Każdorazowo należy rozważyć poziom ochrony, jaki chcemy zapewnić swojemu rozwiązaniu, oraz zakres uprawnień, jakie nadamy użytkownikom końcowym. Niezależnie jednak od wyboru pamiętajmy, żeby jakąś licencję wybrać. W przypadku jej braku, nasze, nawet najbardziej doskonałe, oprogramowanie będzie dla użytkowników całkowicie bezużyteczne. W takiej sytuacji zakres uprawnień wynika wprost z prawa autorskiego i praw pokrewnych i jest dość wąsko zdefiniowany. Nadając licencję, niezależnie od jej charakteru, dajemy możliwość użytkownikowi nabycia praw dodatkowych, które mogą znacznie wpłynąć na użyteczność i popularność naszego rozwiązania, a często umożliwić jego dalszy rozwój nawet wtedy, kiedy my sami nie będziemy w stanie dalej tegoż oprogramowania rozbudowywać.

Bibliografia

1. GNU GENERAL PUBLIC LICENSE, Version 3, 29 June 2007. `https://www.gnu.org/licenses/gpl-3.0.html`, [Online, dostęp: 15.04.2021].
2. The Free Software Foundation (FSF). `https://www.fsf.org/`, [Online, dostęp: 19.04.2021].
3. alavoor: PostgreSQL Licence: GNU/GPL. `https://www.postgresql.org/message-id/20020120210341.90756.qmail%40web10407-.mail.yahoo.com` (2002), [Online, dostęp: 22.04.2021].

4. Bottomley, J.E., Chehab, M.C., Gleixner, T., Hellwig, C., Jones, D., Kroah-Hartman, G., Luck, T., Morton, A., Myklebust, T., Woodhouse, D.: The Dangers and Problems with GPLv3 (2006).
5. Debconf: Linus Torvalds says GPL v3 violates everything that GPLv2 stood for. https://www.youtube.com/watch?v=PaKIZ7gJlRU (2014), [Online, dostęp: 19.04.2021].
6. Free Software Foundation, Inc.: Czemu nie należy stosować Lesser GPL dla kolejnej biblioteki. https://www.gnu.org/licenses/why-not-lgpl.html, [Online, dostęp: 06.05.2021].
7. Free Software Foundation, Inc.: GNU AFFERO GENERAL PUBLIC LICENSE, Version 3, 29 June 2007. https://www.gnu.org/licenses/agpl-3.0.html, [Online, dostęp: 06.05.2021].
8. Free Software Foundation, Inc.: GNU LESSER GENERAL PUBLIC LICENSE, Version 3, 29 June 2007. https://www.gnu.org/licenses/lgpl-3.0.html, [Online, dostęp: 06.05.2021].
9. Free Software Foundation's Licensing and Compliance Lab: Frequently Asked Questions about the GNU Licenses. http://www.gnu.org/licenses/gpl-faq.html#IfInterpreterIsGPL, [Online, dostęp: 05.05.2021].
10. Free Software Foundation's Licensing and Compliance Lab: Rozmaite licencje i komentarze na ich temat#ModifiedBSD. https://www.gnu.org/licenses/license-list.pl.html#ModifiedBSD, [Online, dostęp: 05.05.2021].
11. Gunduz, D.: License clarification: BSD vs MIT. https://www.postgresql-archive.org/License-clarification-BSD-vs-MIT-td2017011.html (2009), [Online, dostęp: 22.04.2021].
12. Kancelaria Sejmu: Ustawa z dnia 4 lutego 1994 r. o prawie autorskim i prawach pokrewnych. In: Dz.U. 1994 nr 24 poz. 83. (1994).
13. Lilly, N.: proposed improvements to PostgreSQL license. https://www.postgresql.org/message-id/3960ACA1.C911664A%40greatbridge.com (2000), [Online, dostęp: 22.04.2021].
14. Open Source Initiative: The 2-Clause BSD License. https://opensource.org/licenses/BSD-2-Clause, [Online, dostęp: 15.04.2021].
15. Open Source Initiative: The 3-Clause BSD License. https://opensource.org/licenses/BSD-3-Clause, [Online, dostęp: 15.04.2021].
16. Open Source Initiative: The MIT License. https://opensource.org/licenses/MIT, [Online, dostęp: 15.04.2021].
17. Open Source Initiative: Zero-Clause BSD (0BSD). https://opensource.org/licenses/0BSD, [Online, dostęp: 23.04.2021].
18. Page, D.: PostgreSQL licence - status? https://web.archive.org/web/201608080930-31/http://www.crynwr.com/cgi-bin/ezmlm-cgi?17:mmp:969 (2010), [Online, dostęp: 22.04.2021].
19. Petreley, N.: A fight against evil or a fight for attention? https://www.linuxjournal.com/content/fight-against-evil-or-fight-attention (2006), [Online, dostęp: 19.04.2021].
20. Schmitz, P.E.: Why viral licensing is a ghost. https://joinup.ec.europa.eu/collection/eupl/news/why-viral-licensing-ghost (2015), [Online, dostęp: 05.05.2021].

21. SPDX: BSD 4-Clause Original or Old License. https://spdx.org/licenses/BSD-4-Clause, [Online, dostęp: 23.04.2021].
22. The Apache Software Foundation: Apache License, version 2.0. https://www.apache.org/licenses/LICENSE-2.0.html, [Online, dostęp: 05.05.2021].
23. The Hermit Hacker: PostgreSQL & the BSD License. https://www.postgresql.org/message-id/Pine.BSF.4.21.0007052208380.33627-100000%40thelab.hub.org (2000), [Online, dostęp: 22.04.2021].
24. The Mozilla Foundation: Mozilla Public License. https://www.mozilla.org/en-US/MPL/, [Online, dostęp: 06.05.2021].
25. The PostgreSQL Global Development Group: PostgreSQL licence. https://www.postgresql.org/about/licence/, [Online, dostęp: 22.04.2021].
26. The SQLite Team: SQLite Is Public Domain. https://www.sqlite.org/copyright.html, [Online, dostęp: 06.05.2021].
27. Wheeler, D.A.: The Free-Libre / Open Source Software (FLOSS) License Slide. https://dwheeler.com/essays/floss-license-slide.html (01 2017), [Online, dostęp: 15.04.2021].
28. Wikipedia contributors: Comparison of free and open-source software licences — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Comparison_of_free_and_open-source_software_licences&oldid=1017649175 (2021), [Online, dostęp: 23.04.2021].