



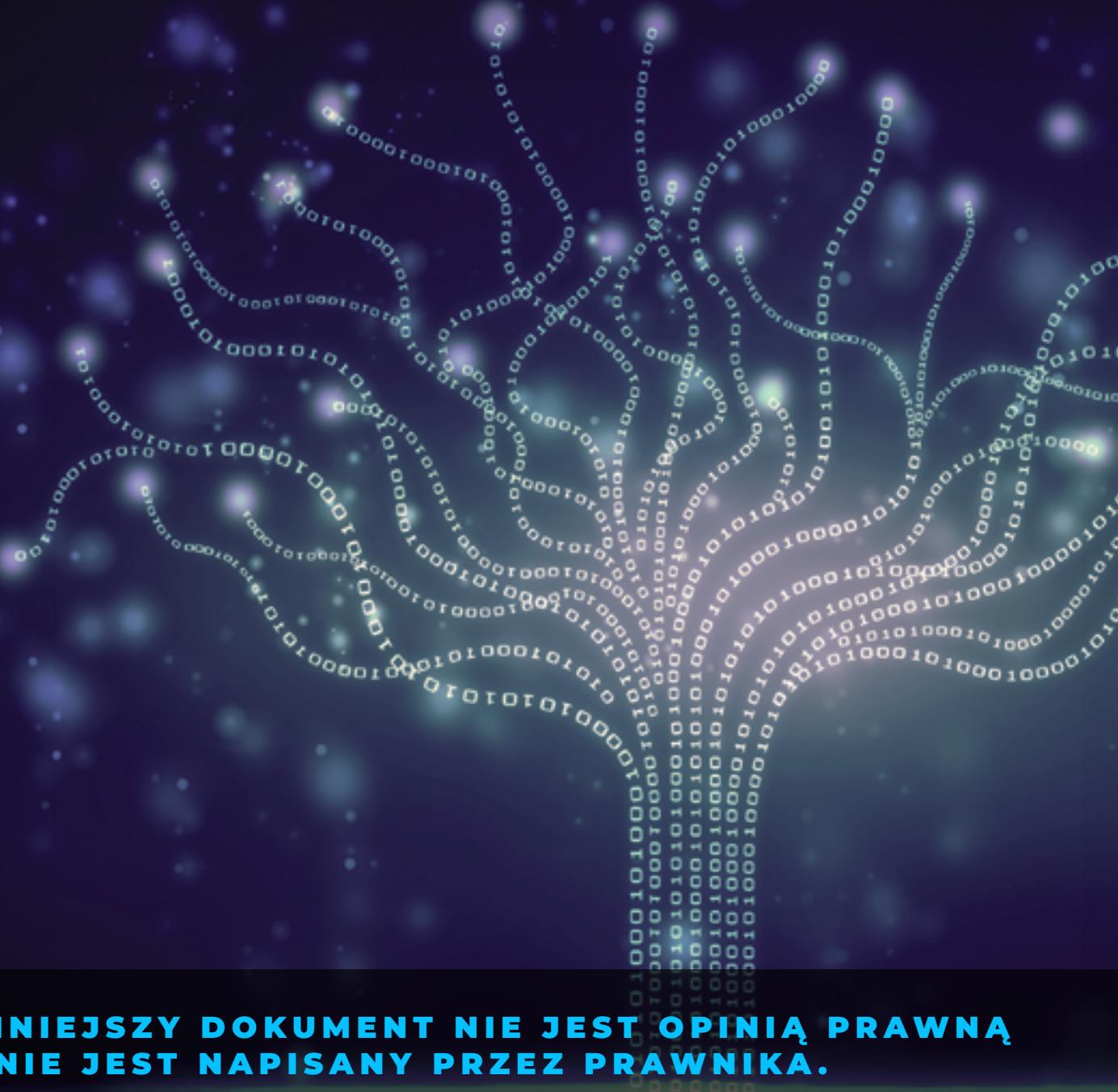
YETI FORCE
CHOOSE OPEN SOURCE

LICENSE

PORADNIK O LICENCJONOWANIU PROJEKTÓW OPEN SOURCE

AUTOR: BŁAŻEJ PABISZCZAK

www.yetiforce.com



**NINIEJSZY DOKUMENT NIE JEST OPINIĄ PRAWNĄ
I NIE JEST NAPISANY PRZEZ PRAWNIKA.**

**JEŻELI CHCESZ WYKORZYSTAĆ WIEDZĘ OPISANĄ W TYM DOKUMENCIE WE
WŁASNEJ ORGANIZACJI, KONIECZNIE SKONSULTUJ JĄ Z PRAWNIKIEM,
KTÓRY SPECJALIZUJE SIĘ W LICENCJONOWANIU ORAZ OPROGRAMOWANIU
OPEN SOURCE.**

04	WPROWADZENIE
05	DLACZEGO TO TAKIE TRUDNE?
07	OD CZEGO NALEŻY ZACZĄĆ?
08	IDENTYFIKACJA LICENCJI W ORGANIZACJI
10	ZABEZPIECZENIE UMOWY
10	ISTOTA PROBLEMU
11	JAK SIĘ ZABEZPIECZAĆ?
12	PORADA DLA STARTUPÓW
13	LICENCJONOWANIE ZŁOŻONYCH PROGRAMÓW
13	GŁÓWNA LICENCJA PROGRAMU
14	DODATKOWE LICENCJE PROGRAMU
15	AUDYT
16	TYPY I KATEGORYZACJA LICENCJI
16	PODZIAŁ ZE WZGLĘDU NA KATEGORIĘ
18	PODZIAŁ ZE WZGLĘDU NA POZIOM RYZYKA
19	ZGODNOŚĆ RÓŻNYCH LICENCJI
20	ŁĄCZENIE LICENCJI W PRAKTYCE
20	MACIERZ ZGODNOŚCI LICENCJI
21	UZUPEŁNIENIA DLA MACIERZY ZGODNOŚCI LICENCJI
22	LISTA BIBLIOTEK KOMPATYBILNYCH Z GPL v2
23	LISTA BIBLIOTEK KOMPATYBILNYCH Z GPL v3
24	LISTA BIBLIOTEK NIEKOMPATYBILNYCH Z GPL v2
26	NAJCZĘŚCIEJ POPEŁNIANE BŁĘDY
26	BAŁAGAN W BIBLIOTEKACH I LICENCJACH
28	ROZPOWSZECHNIANIE LICENCJI Z BIBLIOTEKĄ COPYLEFT
29	ŁĄCZENIE BIBLIOTEK Z LICENCJĄ COPYLEFT
31	PRZEGŁĄD LICENCJI COPYLEFT
31	OSLv3
32	AGPLv3
32	GPLv3
33	PUŁAPKI LICENCYJNE
35	MATERIAŁY ZEWNĘTRZNE

WPROWADZENIE

SŁOWEM WSTĘPU

NA ŚWIETE PANUJE POWSZECHNA NIECHĘĆ DO DZIELENIA SIĘ WIEDZĄ, LUBIMY DZIELIĆ SIĘ RZECZAMI PROSTYMI, BANALNYMI, NIEWYMAGAJĄCYMI OD NAS WIĘKSZEGO ZAANGAŻOWANIA. ŻYJEMY W CZASACH, W KTÓRYCH CHOCЬ DOSTĘP DO WIEDZY JEST POWSZECHNY I ISTNIEJĄ MILIONY STRON Z MILIARDAMI INFORMACJI, TO NADAL MAMY WEWNĘTRZNE POCZUCIE, ŻE INFORMACJE WAŻNE TO TABU.

Tylko osoby w pełni zaangażowane rozumieją, że dzielenie się doświadczeniem, to klucz do lepszego jutra, a nie zagrożenie utraty "niewidzialnej przewagi". Bo jak to możliwe, że w czasach, gdzie 99% firm na świecie korzysta w sposób pośredni lub bezpośredni z otwartego oprogramowania, można znaleźć tak niewiele informacji o jego licencjonowaniu? Gdyby te 99% firm było bardziej świadome, to czy pracy byłoby mniej, czy wręcz na odwrót?

W ostatnich 10 latach brałem udział w realizacji ponad 300 projektów, które wykorzystywały otwarte oprogramowanie. Spośród 300 klientów, tylko jeden z nich (tak, tylko jeden!), zainteresował się szczegółowo licencjonowaniem projektu. Aż ciężko uwierzyć, że tak niewiele firm interesuje się licencjonowaniem, przecież licencja to nic innego, niż umowa pomiędzy stronami. Czyżby firmy nie czytały umów, tak jak informatycy nie czytają instrukcji obsługi?

Gdy zaczynałem swoją przygodę z licencjami open source, nie miałem żadnego doświadczenia, żadnych środków finansowych ani żadnego wsparcia merytorycznego, a chciałem stworzyć ogromny projekt bazujący właśnie na otwartym oprogramowaniu.

Z czego wynika tak duża niechęć firm do poznania i zrozumienia licencji? W praktyce nie kosztuje to przecież zbyt wiele, a inwestowanie w oprogramowanie, które posiada nieznaną nam licencję, jest co najmniej mocno ryzykowne. Tutaj warto nadmienić, że problem nie dotyczy tylko projektów open source, lecz każdego oprogramowania w firmie. Świadomość działów informatycznych i działów prawnych na temat oprogramowania i licencjonowania jest niewielka i w efekcie każda firma boryka się z problemem identyfikacji oprogramowania w swojej firmie.

Kiedyś mój dobry znajomy przy luźnej rozmowie powiedział, że największe wyzwanie w jego firmie (a pracował w firmie zatrudniającej prawie 50 tysięcy pracowników w Polsce) to szczegółowy audit sprzętu i oprogramowania. Przez ponad 8 lat jego pracy na wysokim szczeblu w dziale IT nikt nie podjął się tego wyzwania i nikt nigdy nie podejmował tego tematu przy rozmowach z Zarządem organizacji.

DLACZEGO TO TAKIE TRUDNE?

OD PONAD 6 LAT BARDZO WNIKLIWIE STARAM SIĘ ANALIZOWAĆ, KONSULTOWAĆ I ROZMAWIĄĆ NA TEMAT LICENCJONOWANIA OPEN SOURCE. WYNIKA TO Z DWÓCH CZYNNIKÓW: PIERWSZY TO TAKI, ŻE MUSZĘ TO WIEDZIEĆ JAKO ARCHITEKT SYSTEMU, KTÓRY WYKORZYSTUJE OKOŁO 100 BIBLIOTEK NA 20 RÓŻNYCH LICENCJACH. PO DRUGIE, KONSULTACJE PRAWNE SĄ TAK DROGIE, ŻE KORZYSTAMY Z NICH TYLKO W MOMENTACH, KIEDY UZNAMY TO ZA KONIECZNE.

Gdy już dochodzi do momentu, w którym chcemy skorzystać z doświadczenia prawników, napotykamy kilka problemów:

- ▶ **Pierwszy** to taki, że nie ma zbyt wielu specjalistów w zakresie licencjonowania projektów open source (a przynajmniej ja miałem problem z ich znalezieniem z perspektywy 6 lat i szerokiego grona prawników na LinkedIn). W większości przypadków prawnicy, których znajdowaliśmy, mieli o wiele mniejsze pojęcie w zakresie licencji open source niż my lub specjalizowali się tylko w jednej licencji, np. znali się na GPL, ale nie na AGPL, OSL, MPL itd.
- ▶ **Drugi problem** polega na złożoności zapisów licencji. Niektóre licencje są tak rozbudowane o różne zapisy, że do ich zrozumienia wymagana jest szczegółowa analiza. Z tego powodu wielu ekspertów z dziedziny prawa ma wątpliwości co do interpretacji poszczególnych fragmentów licencji.
- ▶ **Trzeci problem** to koszty opinii prawnych. Bardzo często otrzymywaliśmy wyceny kilkukrotnie przekraczające nasz budżet - np. potrzebowaliśmy opinii prawnej dotyczącej AGPL, do której mieliśmy 10 pytań, a przeznaczony budżet wynosił 10 tysięcy. Po otrzymaniu wycen okazywało się, że potrzeba na to 50-100 tysięcy złotych nawet po obcięciu części pytań.
- ▶ **Czwarty**, najgorszy problem, to brak chęci dzielenia się wiedzą. Jeżeli nawet znajdzie się firma, w której jest prawnik rozumiejący licencje open source, to mało prawdopodobne jest, że jego wiedza i doświadczenie będą publicznie dostępne. O ile jest to zrozumiałe w przypadku prywatnych korporacji, które z założenia są zamknięte na jakiekolwiek dzielenie się wiedzą, o tyle w sektorze publicznym nie powinno się ukrywać wiedzy i doświadczenia.

Uważam, że tutaj najbardziej powinno pomagać państwo, które ma wiedzę, ma środki i ma w tym korzyść, aby podzielić się swoją wiedzą na temat wykorzystywania oprogramowania open source w organizacji. Najwięcej skorzystałoby na tym małe i średnie firmy, których najczęściej nie stać na dodatkowe wydatki prawne oraz na analizę licencji oprogramowania w firmie.



KAŻDA OSOBA W ZESPOLE POWINNA CODZIENNE UCZYĆ SIĘ CZEGOŚ NOWEGO I POWINNA CODZIENNIE POPRAWIAĆ SWOJE NIEDOSKONAŁOŚCI, PONIEWAŻ TO JEDYNA DROGA DO BUDOWANIA SILNEGO I TWÓRCZEGO ZESPOŁU.

OD CZEGO NALEŻY ZACZĄĆ?



W WIĘKSZOŚCI FIRM, Z JAKIMI WSPÓŁPRACOWAŁEM (RÓWNIEŻ JAKO SPECJALISTA IT), NIKT NIE PYTAŁ MNIE O LICENCJE I O TO, JAK Z NIMI POSTĘPOWAĆ. A NAWET JEŻELI KTOŚ BY MNIE SPYTAŁ 10 LAT TEMU, TO CO MIAŁEM ODPOWIEDZIEĆ, NIE ZNAJĄC SIĘ WTEDY NA LICENCJONOWANIU? MOŻNA WIĘC ZAŁOŻYĆ, ŻE TAK NAPRAWDĘ JEST TO TEMAT TABU. JEŻELI FIRMA KUPUJE PROGRAM (WINDOWS, OFFICE, SAP ITP.) LUB POBIERA Z INTERNETU (FIREFOX, LIBREOFFICE, NOTEPAD++ ITP.), TO NIKT NIE BĘDZIE ZAGŁĘBIAŁ SIĘ W LICENCJE, PONIEWAŻ SĄ TO POWSZECHNIE UŻYWANE APLIKACJE.

W niewielkiej firmie takiej jak nasza (aktualnie 20 pracowników), otrzymuję co najmniej jedno zapytanie dziennie dotyczące licencji dla tworzonego lub używanego w organizacji oprogramowania. Gdyby nie szkolenia wewnętrzne dotyczące dozwolonych licencji oraz weryfikacja licencji na każdym etapie wytwarzania oprogramowania, zapewne zapytań nie byłoby w ogóle. Można powiedzieć, że świadomość przychodzi z czasem i jest efektem dojrzałości organizacji.

Aby nie odpowiadać każdemu z osobna i każdorazowo nie analizować tego samego, warto stworzyć dokument/poradnik, który będzie odpowiadał na najczęściej zadawane pytania oraz będzie rósł razem z organizacją. Gdy szukałem takiego dokumentu w polskim sektorze publicznym, niestety nie udało mi się znaleźć żadnego ciekawego materiału. Zacząłem więc szukać w Google i tam natrafiłem na bardzo ciekawy materiał <https://opensource.google.com/docs/>, który opisuje, jak postępować w przypadku oprogramowania na różnych licencjach.

Można z niego wywnioskować kilka fajnych rzeczy, ale najważniejsza jest taka, że użytkownik musitrzymać kompletną wiedzę, jak postępować z nowym oprogramowaniem/projektem w organizacji. Na przykład:

1. Sprawdź, czy program/aplikacja nie była już sprawdzana w naszej bazie i czy nie wydaliśmy już dla niej zaleceń. Bazą w Waszej firmie może być zwykły plik Excela dostępny online lub aplikacja, która będzie do tego dostosowana np. YetiForce CRM.
2. Umieść wszelkie informacje jakie posiadasz, m.in.:
 - nazwę programu/aplikacji,
 - opis/cel,
 - licencję główną,
 - licencje bibliotek zewnętrznych,,
 - inne istotne informacje,
3. Poproś o akceptację osobę uprawnioną (upewnij się, że ma ona dostęp).
4. Po otrzymaniu akceptacji przystęp do pracy.

IDENTYFIKACJA LICENCJI W ORGANIZACJI

KAŻDY PRACOWNIK MUSI ROZUMIEĆ PROCEDURĘ IDENTYFIKACJI I AKCEPTACJI LICENCJI W ORGANIZACJI. JEST TO NIEZALEŻNE OD TEGO CZY MÓWIMY O MAŁYM PROGRAMIE DOSTĘPNYM ONLINE BEZPŁATNIE, CZY TEŻ O DUŻYM OPROGRAMOWANIU WYKORZYSTYWANYM W DZIALE KSIĘGOWOŚCI. JEŻELI W FIRMIE UŻYWA SIĘ OPROGRAMOWANIA, KTÓRE NIE JEST NIGDZIE ZIDENTYFIKOWANE, WÓWCZAS PRĘDZĘ LUB PÓŹNIEJ MOGĄ NAS CZEKAĆ NIEPRZYJEMNOŚCI Z TEGO POWODU (NP.: ODCHODZĄCY PRACOWNIK MOŻE ZGŁOSIĆ NA POLICJĘ, ŻE KORZYSTAMY Z OPROGRAMOWANIA NIEZGODNIE Z LICENCJĄ LUB NIE WYKUPIŁYSMY WYMAGANEJ LICENCJI).

W przypadku oprogramowania, które od początku do końca tworzy jeden producent, najczęściej mamy do czynienia z jedną (główną) licencją, którą musimy zaakceptować - wówczas najczęściej sprawą jest dość prosta, pracownik zgłasza chęć użycia niniejszego programu i prosi o akceptację.

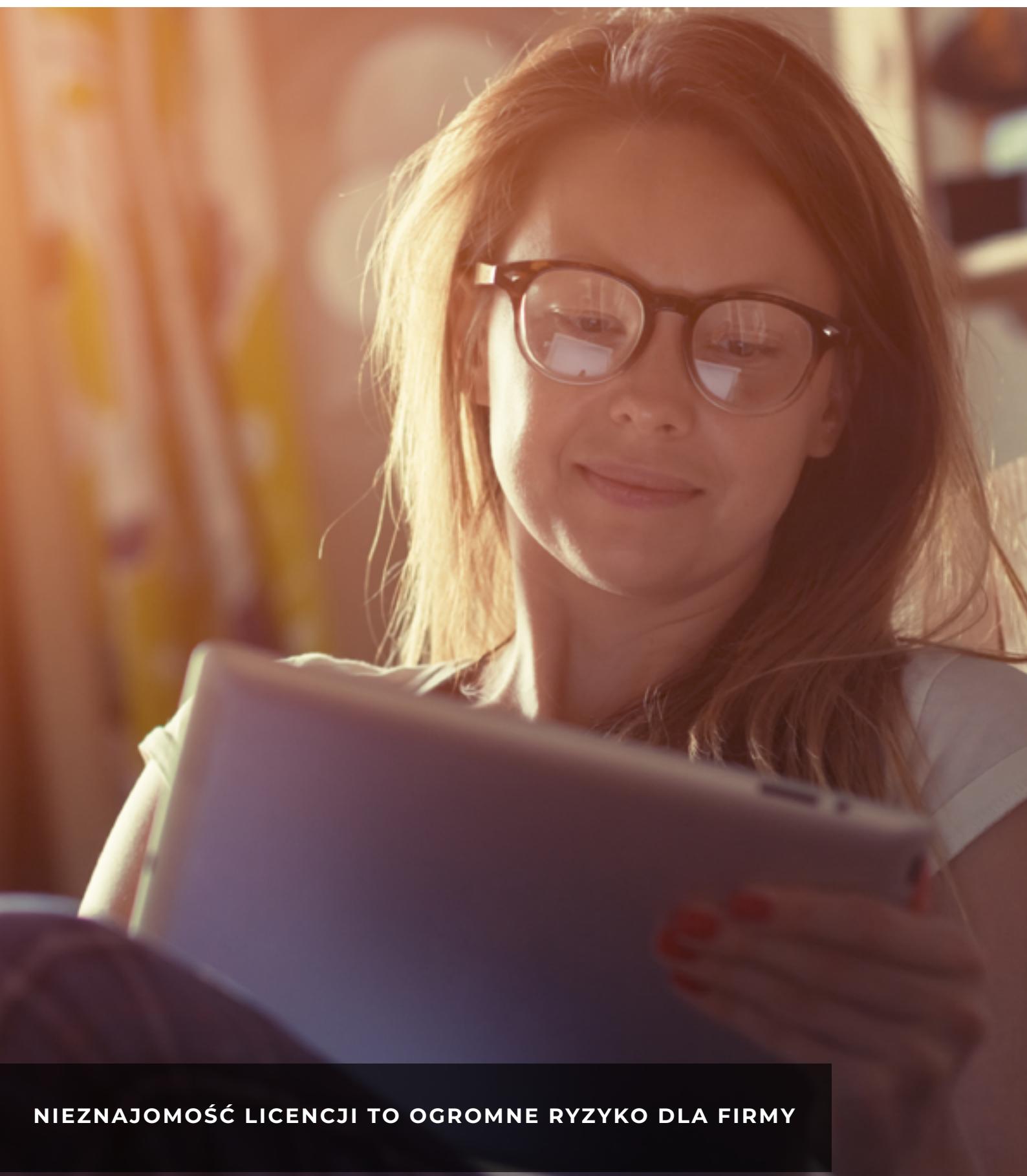
Jeżeli mamy do czynienia z oprogramowaniem złożonym, wówczas, oprócz licencji głównej, musimy akceptować (lub co najmniej mieć świadomość istnienia) wielu bibliotek firm trzecich. W takim przypadku musimy zapoznać się ze wszystkimi bibliotekami i ich licencjami oraz zrozumieć zasady łączenia się licencji, ponieważ w praktyce niektórych licencji nie można łączyć ze sobą (zostało to opisane w oddzielnym punkcie tego poradnika).

Kody niektórych aplikacji bywają chaotyczne i nie zawierają jednej centralnej listy bibliotek firm trzecich i licencji, na jakich są one udostępnione. Gdy na nie natrafimy, powinniśmy zastanowić się, czy nie zrezygnować z ich użytkowania. Jeśli jednak aplikacja jest ważna dla organizacji, należy rozważyć zlecenie audytu takiego oprogramowania.

Porządek w licencjach oznacza dojrzałość oprogramowania/producenta, a co ważne, zmniejsza ryzyko związane z korzystaniem z tej aplikacji. Uważam, że każda firma poszukująca dostawcy oprogramowania, powinna wymagać dostarczenia pełnej listy licencji używanych w aplikacji, aby jeszcze przed wyborem dostawcy zweryfikować, na ile lista jest zgodna z wewnętrznymi zasadami w firmie.

Bardzo często w różnych systemach zdarza się sytuacja, gdy pliki/biblioteki nie posiadają żadnej informacji o licencji. Taki kod powinien być natychmiast albo usunięty z systemu, albo powinno się znaleźć źródło jego pochodzenia i potwierdzić licencję, na jakiej został kod opublikowany. Co najważniejsze, należy sprawdzić, czy taka biblioteka jest na pewno kompatybilna z innymi bibliotekami.

NINIEJSZY DOKUMENT NIE JEST OPINIĄ PRAWNA I NIE JEST NAPISANY PRZEZ PRAWNIKA. JEŻELI CHCESZ WYKORZYSTAĆ WIEDZĘ OPISANĄ W TYM DOKUMENCIE WE WŁASNEJ ORGANIZACJI, KONIECZNIE SKONSULTUJ JĄ Z PRAWNIKIEM, KTÓRY SPECJALIZUJE SIĘ W LICENCJONOWANIU ORAZ OPROGRAMOWANIU OPEN SOURCE.



NIEZNAJOMOŚĆ LICENCJI TO OGROMNE RYZYKO DLA FIRMY

ZABEZPIECZENIE UMOWY

ISTOTA PROBLEMU

“Jeżeli nie stosujemy się do zapisów licencji, wówczas autor chroniony jest na mocy przepisów ustawy z dnia 4 lutego 1994 r. o prawie autorskim i prawach pokrewnych i może szukać ochrony zarówno na gruncie prawa cywilnego jak też prawa karnego. Autor chroniony jest również na podstawie prawa Unii Europejskiej m.in. Dyrektywy Parlamentu Europejskiego i Rady 2009/24/WE z dnia 23 kwietnia 2009 r. w sprawie ochrony prawnej programów komputerowych”

Prawo chroni przede wszystkim autora, dlatego musimy być bardzo uważni, zwłaszcza że:

“Pierwotny licencjonodawca może dochodzić swoich praw na drodze postępowania sądowego od wszystkich podmiotów naruszających jego prawa, nawet jeśli robią to nieświadomie”

Cytat z opinii prawnej pokazuje, jak bardzo złożony jest problem respektowania licencji od strony biznesowej. Jeżeli pobierzemy oprogramowanie (niezależnie, czy jest to oprogramowanie otwarte, czy zamknięte) i to oprogramowanie łamie zapisy licencji, wówczas osoba, która ma pierwotne prawa do tej licencji, może pozwać nas do sądu.

Przykładowy opis sprawy o naruszenie zapisów licencji open source: <https://www.fsf.org/blogs/licensing/update-on-artifex-v-hancom-gnu-gpl-compliance-case-1>.

Tworzenie dobrego oprogramowania często zajmuje lata, co powoduje, że w przypadku wielu aplikacji tworzonych przez “naście” lat czasem nie wiadomo, skąd pochodzi kod i jaka jest jego pierwotna licencja. Oczywiście w dojrzałej organizacji, która jest producentem, nie powinno to mieć miejsca.

Dodatkowo, zawsze należy brać pod uwagę możliwość ryzyka, że programista użył niedozwolonej biblioteki i ukrył informacje o autorze. Dlatego tak ważny jest zewnętrzny audit lub statyczna analiza kodu. Jakość musi być procesem i na każdym etapie projektowania i wytwarzania oprogramowania należy zwracać szczególną uwagę na licencjonowanie.

“Aż 74% skontrolowanych projektów zawierało elementy niezgodne z główną licencją projektu, w tym aż 44% dotyczyło licencji GPL.”

Źródło: 2018 Open Source Security and Risk Analysis - Synopsys

JAK SIĘ ZABEZPIECZAĆ?

WAŻNE JEST ZABEZPIECZENIE SIĘ PRZED MOŻLIWYMI SKUTKAMI PRAWNYMI DOCHODZENIA PRZEZ KOGOŚ JEGO PRAW DO OPROGRAMOWANIA, KTÓREGO MY UŻYWAMY. NAJLEPSZYM SPOSOBEM JEST STOSOWANIE ODPOWIEDNICH ZAPISÓW POMIĘDZY STRONAMI, PONIŻEJ PRZYKŁAD, KTÓRY SAMI STOSUJEMY U SIEBIE DLA NASZYM Klientów:

- 1.** Strony potwierdzają, że Zamawiający jest uprawniony do korzystania z Oprogramowania [dalej "Oprogramowanie"] YetiForce udostępnionego przez Dostawcę, w tym jego wszelkich modyfikacji, zgodnie z zasadami licencji publicznej dostępnej pod adresem: "<https://github.com/YetiForceCompany/YetiForceCRM/tree/developer/licenses>".
- 2.** Dostawca potwierdza, iż zgodnie z zasadami udzielanej przez niego licencji publicznej, o której mowa w ust. 1 powyżej, w żadnym wypadku i na żadnej podstawie prawnej Zamawiający nie ponosi odpowiedzialności wobec jakiegokolwiek podmiotu trzeciego za jakiekolwiek szkody (pośrednie i bezpośrednie), związane z korzystaniem z Oprogramowania.
- 3.** W przypadku roszczeń zgłoszonych przez osoby trzecie wskazujących, że Oprogramowanie dostarczone przez Dostawcę narusza prawa własności intelektualnej tych osób, Dostawca będzie na własny koszt bronić Zamawiającego przed takimi roszczeniami oraz pokryje wszelkie koszty, odszkodowania i koszty obsługi prawnej, zasądzone ostatecznie przez sąd lub zawarte w ugodzie zatwierdzonej przez Dostawcę, pod warunkiem, że Zamawiający:
 - ▶ powiadomi niezwłocznie na piśmie Dostawcę o takim roszczaniu,
 - ▶ pozwoli Dostawcy na przejęcie kontroli i będzie z nim współpracował w zakresie obrony przed takimi roszczeniami oraz w zakresie prowadzenia jakichkolwiek negocjacji w celu ugodowego rozstrzygnięcia sporu.
- 4.** Jeżeli Oprogramowanie zostanie uznane za naruszające prawa własności intelektualnej lub prawa autorskie osoby trzeciej, Dostawca na swój koszt i według własnego wyboru uzyska dla Zamawiającego prawa do korzystania z Oprogramowania na warunkach opisanych w umowie licencji publicznej lub dokona wymiany Oprogramowania na inne nie naruszające praw, o których mowa w ust. 3 powyżej, w taki sposób aby Zamawiający nieprzerwanie korzystał z Oprogramowania lub innego oprogramowania spełniającego te same funkcje i wymagania, co Oprogramowanie.
- 5.** Dla uniknięcia wątpliwości, Dostawca nie ponosi odpowiedzialności z tytułu roszczeń, których przedmiotem jest jakikolwiek kod dołączony przez Zamawiającego.

PORADA DLA STARTUPÓW



PIERWSZE PODJĘTE DECYZJE MOGĄ BYĆ KLUCZOWE

W 2013 ROKU JAKO FIRMA, KTÓRA CHCIAŁA STWORZYĆ FORK SYSTEMU OPEN SOURCE, ZASTANAWIAŁIŚMY SIĘ NAD SYSTEMEM, KTÓRY MOŻEMY WYKORZYSTAĆ. GŁÓWNY PROBLEM, JAKI MUSIELIŚMY ROZWIĄZAĆ, TO LICENCJONOWANIE SYSTEMÓW, KTÓRE BRAŁIŚMY POD UWAGĘ ORAZ JAKOŚĆ KODU. DECYZJA MIAŁA ZAPAŚĆ POMIĘDZY DWOMA SYSTEMAMI: SUGARCE I VTIGERCRM.

Choć firma była niewielka i nie miała środków na opinię prawną, to zleciliśmy niezależną analizę dla tych dwóch systemów, wydając na to około 10.000 zł (była to niewielka suma, patrząc z perspektywy dnia dzisiejszego, jednakże wówczas było to wszystko, co mieliśmy i mogliśmy wydać). Ta opinia prawa utwierdziła nas, że dokonaliśmy dobrego wyboru i z perspektywy lat możemy angażować i środki i ludzi w rozwój systemu pod własną marką. Co ważne, opinię realizowała kancelaria specjalizująca się właśnie w licencjach open source.

Gdybyśmy wówczas wybraли SugarCE, to najprawdopodobniej nie moglibyśmy brać udziału w większości przetargów (ze względu na licencję) i większość klientów, którzy dowiedzieliby się o konsekwencjach korzystania z licencji AGPL, przestałby dalej z nami rozmawiać.

Zła decyzja podjęta wówczas powodowałaby milionowe straty dzisiaj i ograniczałaby znacząco rozwój spółki. Należy pamiętać, że jako firma jesteśmy odpowiedzialni nie tylko za siebie, ale również za naszych klientów.

Obecnie nasz projekt YetiForce jest jednym z najlepiej uporządkowanych systemów klasy CRM, jeżeli chodzi o biblioteki i licencjonowanie. W każdej chwili administrator systemu może nie tylko zweryfikować listę bibliotek, ale również wersję biblioteki oraz jej licencje. W wielu przypadkach użytkownik ma dostęp do licencji w wersji angielskiej, ale również w wersji polskiej. Najnowsza wersja systemu została opublikowana na licencji **YetiForce Public License 3**, która jest rozszerzoną licencją MIT.

LICENCJONOWANIE ZŁOŻONYCH PROGRAMÓW

GŁÓWNA LICENCJA PROGRAMU

NALEŻY PAMIĘTAĆ, ŻE LICENCJA, JAKA BY NIE BYŁA, MA NA CELU OGRANICZANIE NASZYCH PRAW WZGLĘDEM UTWORU, KTÓREGO DOTYCZY. JEŻELI JEDNAK KTÓŚ JUŻ STWORZYŁ LICENCJĘ DO OPROGRAMOWANIA, TO TYLKO PO TO, ABY NAS POINFORMOWAĆ, CO MOŻEMY, CZEGO NIE MOŻEMY, A CO MUSIMY. LICENCJA JEST JEDNOCZEŚNIE UMOWĄ POMIĘDZY STRONAMI, A WIĘC DOBRZE, ABYŚMY ZNALI TREŚĆ LICENCJI (UMOWY), BY NIE ŁAMAĆ JEJ WARUNKÓW.

Aktualnie istnieją setki licencji open source, a samych licencji, które zostały zaakceptowane przez Open Source Initiative, jest ponad 80. W wielu przypadkach licencje open source są łagodniejsze niż w przypadku licencji dotyczących oprogramowania zamkniętego, ale nie jest to regułą.

Istnieją strony, które zawierają listę licencji open source, np.: <https://opensource.org/licenses/alphabetical>, niestety większość z nich jest dostępna tylko w języku angielskim. Jeżeli nawet znajdziemy którąś licencję w języku polskim, to najczęściej obowiązującą jest wersja w języku angielskim. W mojej opinii brak dostępu do licencji w języku polskim powoduje w naszym kraju tak wielką ignorancję, jeżeli chodzi o ich czytanie.

Dodatkowo możemy znaleźć wiele miejsc, które pomagają zrozumieć poszczególne licencje

za pomocą podsumowania i uporządkowania w trzech kategoriach: co możesz, czego nie możesz oraz co musisz: <https://tldrlegal.com/>.

Jeżeli nasze oprogramowanie zawiera licencję open source (np. MIT), a jednocześnie nie mamy prawnika specjalizującego się w licencjonowaniu otwartych aplikacji, spróbujmy dopasować licencję w narzędziu podanym powyżej: <https://tldrlegal.com/license/mit-license#summary>. W ten sposób otrzymamy najważniejsze informacje dotyczące tej licencji.

W każdym przypadku należy zachować ostrożność, ponieważ może się okazać, że sama aplikacja może być licencjonowana inaczej, niż nam lub producentowi się wydaje, jeżeli zostanie ona połączona z inną licencją (np. z licencją typu copyleft). Dlatego równie ważne są dla nas biblioteki firm trzecich dołączone do oprogramowania.

"Aby odnieść sukces na dzisiejszym rynku, firmy muszą wprowadzać innowacje! Tempo innowacji wyklucza budowanie wszystkiego od zera. Szacuje się, że nowoczesne aplikacje zawierają od 80-90% kodu open source!"

Źródło: [forbes.com](https://www.forbes.com/sites/forbestechcouncil/2018/02/06/the-most-common-open-source-licensing-mistakes-and-how-to-fix-them/#:~:text=The%20most%20common%20open%20source%20licensing,mistakes%20and%20how%20to%20fix%20them)

DODATKOWE LICENCJE PROGRAMU

DODATKOWE LICENCJE PROGRAMU TO LICENCJE BIBLIOTEK FIRM TRZECICH, KTÓRE NAJCZĘŚCIEJ ZNAJDUJĄ SIĘ W FOLDERZE “THIRD_PARTY” LUB “VENDOR”. CO CIEKAWE, LICENCJE TE SĄ RÓWNIE WAŻNE, CO LICENCJA GŁÓWNA PROGRAMU. MOŻE SIĘ ZDARZYĆ, ŻE LICENCJE DODATKOWE ODDZIAŁUJĄ NA LICENCJĘ GŁÓWNA, DLATEGO WAŻNE JEST ZAPOZNANIE SIĘ ZE WSZYSTKIMI LICENCJAMI W APLIKACJI.

Każde oprogramowanie powinno mieć uporządkowane kwestie bibliotek i ich licencji. Jeżeli tak nie jest, zalecamy albo porzucić to oprogramowanie albo wykonać jego audit.

Jeżeli mamy do czynienia z dojrzałym oprogramowaniem, otrzymamy pełne zestawienie wszystkich bibliotek (razem z licencjami i numerami wersji) w czytelnej formie z możliwością sortowania i zapoznania się z każdą licencją. Poniżej zrzut ekranu z systemu YetiForce:

Nazwa	Wersja	Licencja	Akcje	
yetiforce	4.4	YetiForce Public License v3		
Viger (Orginalnego kodu Vigena zostało około 10%. Przesłany kod został przepisany i rozszerszony)	6.4.0 rev. 14548	VPL 1.1		
Sugar CRM (Viger bazował na Sugar CRM. Obecnie w YetiForce pozostało kilka plików po Sugar CRM)		SPR-1.1.2		
akredytuj/akreditor	4.10.0	MPL-1.1+ (GPL-2.0+, LGPL-2.1+, MPL-1.1+)		
composer/tauris	1.1.1	MIT		
composer/tauris-bundle	v1.3	BSD-3-Clause		
digibio/php	v1.3	MIT		
doctrine/collections	v1.5.0	MIT		
doctrine/inflector	v1.3.0	MIT		
ezymq/ezymq/poller	v4.19.0	AGPL		
guzzlehttp/phone-number-bin.php	0.9.11	Apache-2.0		
guzzlehttp/locale	1.5	MIT		
illuminate/contracts	v5.6.29	MIT		
illuminate/support	v5.6.29	MIT		
keythysys/lip-can-parser	1.4.1	MIT		
monolog/monolog	1.3.6	AGPL-3.0		
nesbot/carbon	1.32.0	MIT		
phpmailer/phpmailer	v6.0.5	AGPL-2.1		
phppot/php-potraitheet	1.3.1	AGPL-2.1-or-later		
psr/container	1.0.0	MIT		
psr/log	1.0.2	MIT		
psr/simple-cache	1.0.1	MIT		
rmonaut/requiert	v1.7.0	ISC		
sabre/dav	3.2.2	BSD-3-Clause		
sabre/event	3.6.0	BSD-3-Clause		
sabre/http	v4.2.4	BSD-3-Clause		
sabre/uri	3.2.1	BSD-3-Clause		
sabre/voicet	4.1.6	BSD-3-Clause		
sabre/xml	3.5.0	BSD-3-Clause		
sabre/dav/tarantula_chekker	v4.1.8	MIT		

Jak widać na zrzucie ekranu powyżej, mamy pełną listę bibliotek razem z numerem wersji, nazwą licencji z możliwością pobrania jej treści i podglądu jej szczegółów. Jeżeli biblioteka jest publikowana pod różnymi licencjami, wówczas widać oznaczenie, która licencja została wybrana.

Producent SugarCRM również udostępnia dla wersji komercyjnych swojego produktu informacje o bibliotekach i licencjach w oddzielnym dokumencie PDF:

<https://support.sugarcrm.com/files/d38fd368-35dd-11e7-93de-06ac51574995.pdf>

AUDYT

MOŻE SIĘ ZDARZYĆ, ŻE TRAFI NAM SIĘ OPROGRAMOWANIE, KTÓREGO LISTA BIBLIOTEK I ICH LICENCJI NIE JEST UPORZĄDKOWANA. JEŻELI NIE CHCEMY ZREZYGNOWAĆ Z TAKIEGO OPROGRAMOWANIA, TO SPRÓBUJMY WYKONAĆ JEGO AUDYT. W SYTUACJI, GDY MAMY DO CZYNNIENIA Z OPROGRAMOWANIEM OPEN SOURCE, TO NIE MA NAJMIEJSZEGO PROBLEMU, ABY TAKI AUDYT WYKONAĆ (MOŻEMY TO ZROBIĆ NAWET SAMI).

Jeżeli jest to oprogramowanie zamknięte, wówczas audyt powinna wykonać firma niezależna od producenta/dostawcy i choć nie będziemy mieli dostępu do kodu źródłowego, powinniśmy otrzymać dostęp do raportu z audytu. Jeżeli i to nie jest możliwe, wówczas niestety odradzamy korzystanie z tego typu oprogramowania.

Audyt możemy wykonać albo ręcznie (wyznaczając osobę, która będzie wchodzić w każdy folder i plik aplikacji, a następnie będzie zapisywać w arkuszu wszystkie znalezione biblioteki i licencje), albo automatycznie, wykorzystując gotowe narzędzia, np. <https://fossa.io>.

The screenshot shows the Fossa interface with the following data:

Project	Status	Dependencies	Licenses	Last Updated
EPESI	5 Issues Found	6 deps	21 licenses	6 days ago
SuiteCRM	3 Issues Found	37 deps	22 licenses	6 days ago
VtigerCRM	2 Issues Found	83 deps	34 licenses	6 days ago
YetiForceCRM	No Issues Found	29 deps	21 licenses	6 days ago

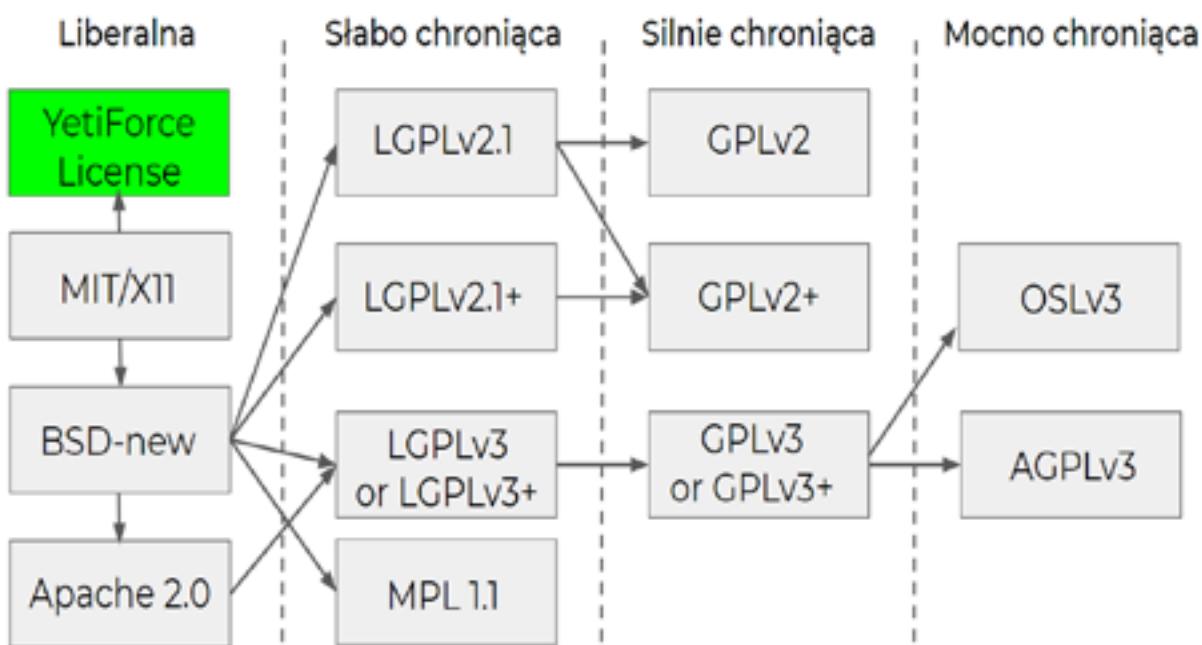
At the bottom, there is a footer note: "Open source management powered by FOSSA, Inc. © 2018 v0.7.0 - revid: n/a".

Jak widać na obrazku powyżej, Fossa znalazła kilka problemów dla aplikacji EPESI, SuiteCRM i VtigerCRM. Każdy z problemów należy przeanalizować ręcznie, ponieważ Fossa może źle zinterpretować niektóre licencje, w szczególności gdy biblioteka jest udostępniana na kilku licencjach jednocześnie.

Niestety większość systemów biznesowych (w szczególności tych mniej popularnych) ma problemy z licencjami, na przykład: SuiteCRM, VtigerCRM, coreBOS, joForce, EPESI, EspoCRM, Dolibarr. Niestety automatyczne narzędzia nie znajdują poprawnie wielu innych problemów, które można znaleźć przy audycie ręcznym, ale więcej na ten temat w dalszej części poradnika.

TYPY I KATEGORYZACJA LICENCJI

PODZIAŁ ZE WZGLĘDU NA KATEGORIĘ



01 LIBERALNA

Nie wymusza na nas stosowania tej samej licencji w naszym kodzie, należy przede wszystkim zachować informację o licencji oraz informacje w pliku (to tylko uproszczenie biznesowe, aby ułatwić zrozumienie różnic pomiędzy różnymi kategoriami licencji).

02 SŁABO CHRONIĄCA

Nie wymusza na nas stosowania tej samej licencji w naszym kodzie, o ile kod ten nie jest częścią biblioteki, a sama biblioteka nie jest podstawową funkcjonalnością aplikacji. Np. jeżeli nasz system spośród wielu funkcjonalności chce do wydruku PDF użyć biblioteki na licencji np. LGPL, a pozostała część aplikacji jest na innej licencji, wówczas możemy użyć licencji LGPL (to tylko uproszczenie biznesowe, aby ułatwić zrozumienie różnic pomiędzy różnymi kategoriami licencji).

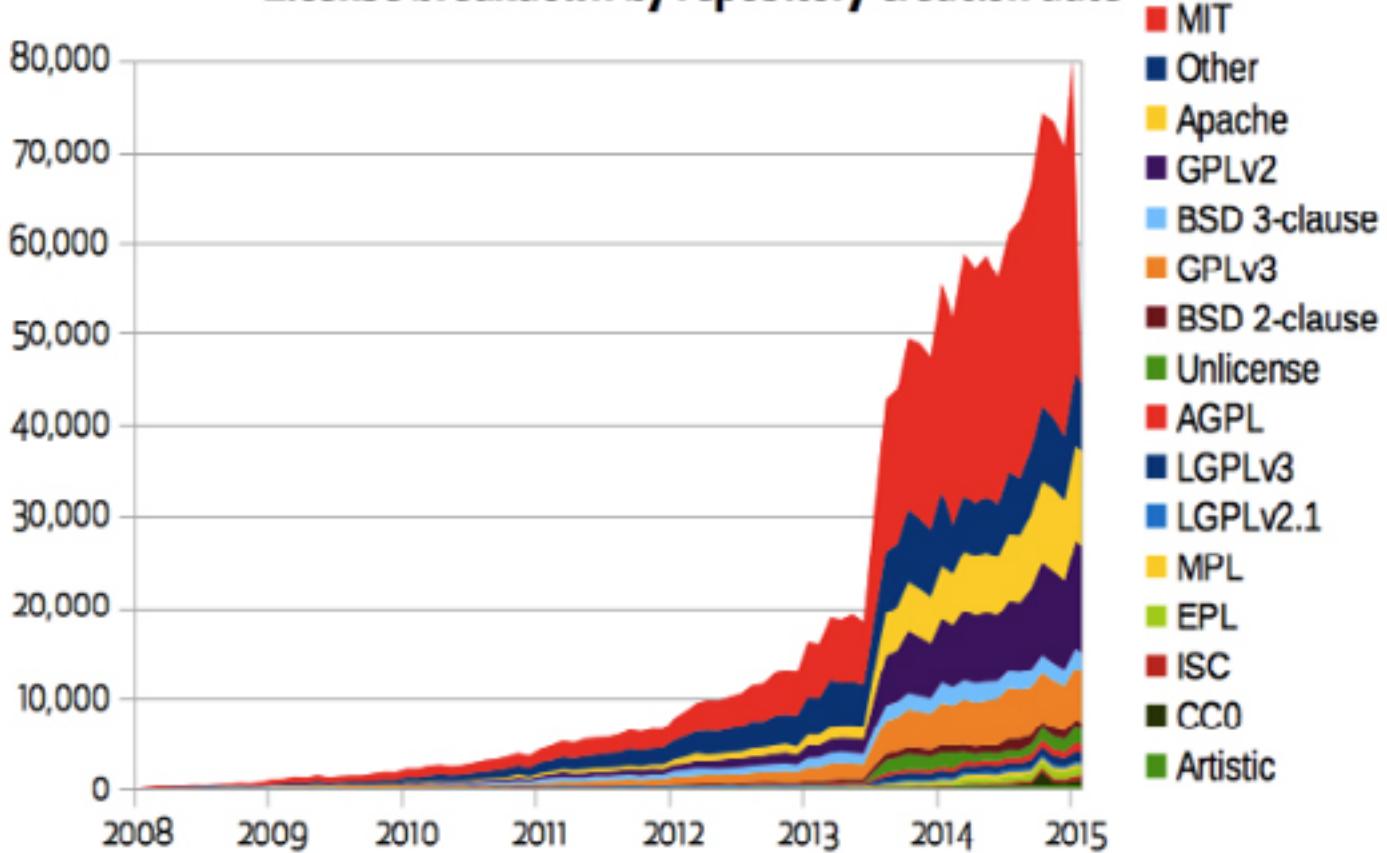
03 SILNIE CHRONIĄCA

Wymusza na nas, aby cała aplikacja była na tej samej licencji, jeżeli choć mały fragment kodu zostanie użyty w aplikacji i rozpowszechniany razem z nią (to tylko uproszczenie biznesowe, aby ułatwić zrozumienie różnic pomiędzy różnymi kategoriami licencji).

04 MOCNO CHRONIĄCA

Podobnie jak w przypadku kategorii licencji silnie chroniących, z taką różnicą, że do rozpowszechnienia dochodzi już albo na etapie dostępu online do aplikacji albo przy wgrywaniu zmian do niej (to tylko uproszczenie biznesowe, aby ułatwić zrozumienie różnic pomiędzy różnymi kategoriami licencji).

License breakdown by repository creation date



Źródło: <https://blog.github.com/2015-03-09-open-source-license-usage-on-github-com/>

PODZIAŁ ZE WZGLĘDU NA POZIOM RYZYKA

WARTO, ABY FIRMA POSIADAŁA “WSTĘPNĄ” LISTĘ LICENCJI I SPOSOBÓW POSTĘPOWANIA NA WYPADEK KONIECZNOŚCI WYBORU OPROGRAMOWANIA. PONIŻEJ ZNAJDUJE SIĘ NASZA WEWNĘTRZNA LISTA, KTÓRĄ CAŁY CZAS ROZSZERZAMY I WERYFIKUJEMY.

NISKIE RYZYKO	UMIARKOWANE RYZYKO	WYSOKIE RYZYKO
Oprogramowanie może być używane w organizacji i nie wymaga każdorazowej akceptacji, o ile źródło pochodzenia aplikacji jest bezpieczne i nie jest potrzebna szersza analiza wewnętrzna	Oprogramowanie stwarza dodatkowe ryzyko, które może zależeć od sposobu, w jaki oprogramowanie jest używane lub od sposobu, w jaki wykorzystuje je programista. Wymaga każdorazowo zgody wewnętrz firm na korzystanie z biblioteki.	Oprogramowanie stwarza wysokie ryzyko ze względu na rodzaj licencjonowania copyleft, gdzie każdy nowy kod musi być udostępniany na tej samej licencji. Wymaga każdorazowo zgody wewnętrz firm na korzystanie z biblioteki.
<input checked="" type="checkbox"/> Apache 1.0, 1.1 <input checked="" type="checkbox"/> Apache 2.0 <input checked="" type="checkbox"/> BSD (all versions) <input checked="" type="checkbox"/> Internet Software Consortium (ISC) <input checked="" type="checkbox"/> JDOM license <input checked="" type="checkbox"/> MIT (all versions) <input checked="" type="checkbox"/> Python Software Foundation license <input checked="" type="checkbox"/> YetiForce Public License 3 <input checked="" type="checkbox"/> WTFPL	<input checked="" type="checkbox"/> Apple Public source license <input checked="" type="checkbox"/> Artistic License 1.0/2.0 <input checked="" type="checkbox"/> Code Project Open license (all versions) <input checked="" type="checkbox"/> Common Development and Distribution (CDDL) 1.0/1.1 <input checked="" type="checkbox"/> Common Public Attribution License (CPAL) 1.0 <input checked="" type="checkbox"/> Commons Public License v1 <input checked="" type="checkbox"/> Eclipse Public License v1 license <input checked="" type="checkbox"/> IBM Public License v1 <input checked="" type="checkbox"/> Microsoft Public License v1 <input checked="" type="checkbox"/> Mozilla Public License v1.1 <input checked="" type="checkbox"/> Mozilla Public License v2 license <input checked="" type="checkbox"/> SVNkit license <input checked="" type="checkbox"/> Sugar Public License 1.1.2 <input checked="" type="checkbox"/> Vtiger Public License 1.1	<input checked="" type="checkbox"/> Affero GNU Public License v3 <input checked="" type="checkbox"/> Less GNU Public License v2 license <input checked="" type="checkbox"/> Less GNU Public License v3 <input checked="" type="checkbox"/> GNU Public License v2 <input checked="" type="checkbox"/> GNU Public License v3 <input checked="" type="checkbox"/> Open Software License (OSL) All versions license

Należy pamiętać, że ryzyko wykorzystania licencji może zależeć m.in. od tego, w jaki sposób wykorzystujemy oprogramowanie. Przykładowo, jeżeli wybierzymy oprogramowanie na licencji LGPL i będziemy je rozbudowywać, wówczas wszystko, co dodamy, musi również być na licencji LGPL. Jeżeli to samo oprogramowanie na licencji LGPL dodamy jako bibliotekę i będzie to tylko niewielki fragment naszego głównego oprogramowania, wówczas nasze oprogramowanie nie musi być na licencji LGPL (to tylko uproszczenie, aby móc lepiej zrozumieć ogólną zasadę, każdy taki przypadek wymaga każdorazowo analizy prawnej).

ZGODNOŚĆ RÓŻNYCH LICENCJI



PROBLEM POJAWIA SIĘ WÓWCZAS, GDY MAMY W SYSTEMIE KILKADZIESIĄT RÓŻNYCH LICENCJI I MUSZĄ BYĆ ONE ZE SOBĄ ZGODNE, ABY SIĘ NIE WYKLUCAŁY. SĄ LICENCJE, KTÓRE SĄ BARDZO LIBERALNE I NIE STANOWIĄ PRAKTYCZNIE ŻADNYCH PROBLEMÓW ZARÓWNO OD STRONY PRAWNEJ, JAK RÓWNIEŻ OD STRONY BIZNESOWEJ, ALE SĄ RÓWNIEŻ LICENCJE, NA KTÓRE NALEŻY BARDZO UWAŻAĆ, PONIEWAŻ SĄ BARDZO RESTRYKCYJNE.

Każdy producent oprogramowania musi mieć uporządkowane kwestie licencjonowania już na samym początku projektu, tak aby przez cały czas mieć pełną kontrolę nad kodem w swojej aplikacji. Jeżeli rozmawiasz z dostawcą oprogramowania, koniecznie poproś go o listę wszystkich bibliotek i licencji, jakie są użyte w oprogramowaniu.

Dodatkowo możemy posługiwać się narzędziami, które zostały zaprojektowane do analizowania kodu i wyszukiwania w nim wszystkich bibliotek. Dobrym przykładem takiego narzędzia jest <https://fossa.io/>, które dla projektów open source jest bezpłatne.

Oprócz standardowej listy licencji użytych w oprogramowaniu powinniśmy szukać tzw. zależności pomiędzy bibliotekami, jak również

konfliktów licencyjnych, ponieważ różne wersje licencji są lub nie są kompatybilne ze sobą. Przykładowo licencja AGPL w wersji 2 nie jest kompatybilna z licencją GPL w wersji 2, natomiast AGPL i GPL w wersji 3 są już ze sobą kompatybilne.

Większość licencji copyleft bazuje na licencji GPL [General Public License]. Osobiście zalecam korzystanie z wersji 3 tejże licencji, istnieją jednak projekty, która uważają wersję 2 licencji za wystarczającą (<https://www.cnet.com/news/torvalds-no-gpl-3-for-linux/>). Główna przewaga licencji w trzeciej wersji została opisana m.in. tutaj: <http://www.ifross.org/en/what-difference-between-gplv2-and-gplv3>, a co najważniejsze, wersja ta umożliwia łączenie z większą liczbą innych licencji, np. AGPL v3 i Apache License v 2.0.

ŁĄCZENIE LICENCJI W PRAKTYCE

MACIERZ ZGODNOŚCI LICENCJI

RÓŻNE LICENCJE GNU CIESZĄ SIĘ KOMPATYBILNOŚCIĄ MIĘDZY SOBĄ - Z JEDNYM WYJĄTKIEM. NIE MOŻNA POŁĄCZYĆ KODU O STARSZEJ WERSJI Z KODEM W NOWSZEJ WERSJI. PONIŻEJ ZNAJDUJE SIĘ SZCZEGÓŁOWA STRUKTURA KOMPATYBILNOŚCI DLA RÓŻNYCH KOMBINACJI LICENCJI GNU. STRUKTURA TA WSKAZUJE, CZY MOŻESZ WYKORZYSTAĆ CZYJŚ KOD NA DANEJ LICENCJI, LICENCJONUJĄC JEDNOCZEŚNIE SWÓJ KOD POD INNĄ.

		I want to license my code under:					
		GPLv2 only	GPLv2 or later	GPLv3 or later	LGPLv2.1 only	LGPLv2.1 or later	LGPLv3 or later
I want to copy code under:	GPLv2 only	OK	OK [2]	NO	OK: Combination is under GPLv2 only [2]	OK: Combination is under GPLv2 only [2][3]	NO
	GPLv2 or later	OK [1]	OK	OK	OK: Combination is under GPLv2 or later [2]	OK: Combination is under GPLv2 or later [2]	OK: Combination is under GPLv3 [8]
	GPLv3	NO	OK: Combination is under GPLv3 [3]	OK	OK: Combination is under GPLv3 [2]	OK: Combination is under GPLv3 [2]	OK: Combination is under GPLv3 [8]
	LGPLv2.1 only	OK: Convey copied code under GPLv2 [2]	OK: Convey copied code under GPLv2 or later [2]	OK: Convey copied code under GPLv3 [2]	OK	OK [6]	OK: Convey copied code under GPLv3 [2][8]
	LGPLv2.1 or later	OK: Convey copied code under GPLv2 [2][1]	OK: Convey copied code under GPLv2 or later [2]	OK: Convey code under GPLv3 [2]	OK [5]	OK	OK
I want to use a library under:	LGPLv3	NO	OK: Combination is under GPLv3 [8][3]	OK: Combination is under GPLv3 [8]	OK: Combination is under GPLv3 [2][8]	OK: Combination is under LGPLv3 [4]	OK
	GPLv2 only	OK	OK [2]	NO	OK: Combination is under GPLv2 only [2]	OK: Combination is under GPLv2 only [2][3]	NO
	GPLv2 or later	OK [1]	OK	OK	OK: Combination is under GPLv2 or later [2]	OK: Combination is under GPLv2 or later [2]	OK: Combination is under GPLv3 [8]
	GPLv3	NO	OK: Combination is under GPLv3 [3]	OK	OK: Combination is under GPLv3 [2]	OK: Combination is under GPLv3 [2]	OK: Combination is under GPLv3 [8]
	LGPLv2.1 only	OK	OK	OK	OK	OK	OK
	LGPLv2.1 or later	OK	OK	OK	OK	OK	OK
	LGPLv3	NO	OK: Combination is under GPLv3 [9]	OK	OK	OK	OK

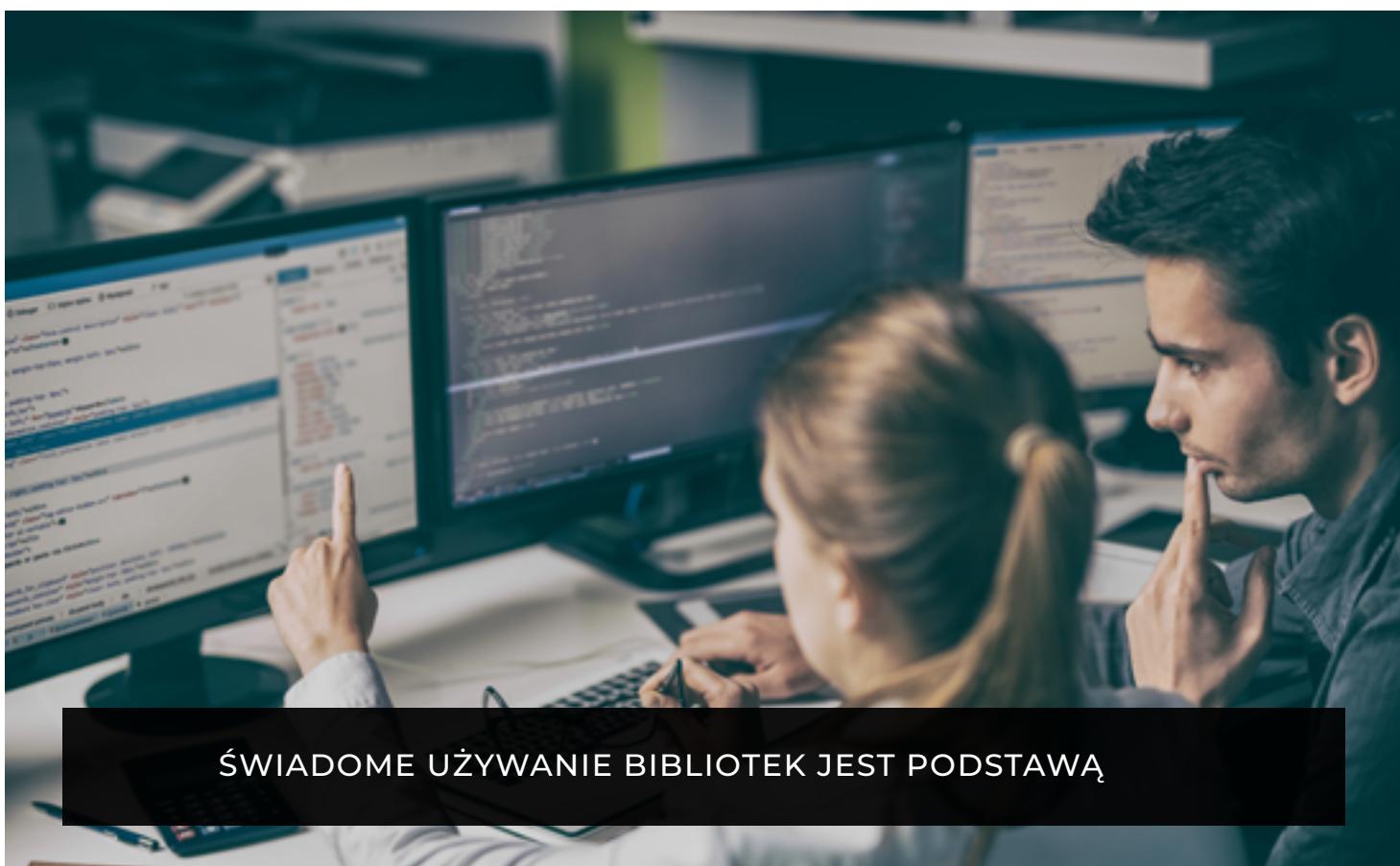
Źródło: <https://www.gnu.org/licenses/gpl-faq.en.html#AllCompatibility>

Znajdź licencję dla swojego projektu w kolumnie u góry tabeli, a licencję dla drugiego kodu w rzędzie po lewej stronie. Komórka, w której się spotkają, powie ci, czy ta kombinacja jest dozwolona. Mówiąc "kopiować kod", mamy na myśli to: bierzesz fragment kodu z jednego źródła, modyfikujesz go lub nie i wstawiasz do własnego oprogramowania, tworząc w ten

sposób dzieło na podstawie pierwszej sekcji kodu. "Użyć biblioteki" oznacza, że nie kopujesz bezpośrednio żadnego źródła, ale zamiast tego wchodzisz z nim w interakcję poprzez łączenie, importowanie, lub inne typowe mechanizmy, które łączą źródła razem podczas kompilowania, lub wykonywania kodu. Wszędzie gdzie, widzisz GPLv3, można również dobrze wstawić AGPLv3.

UZUPEŁNIENIA DLA MACIERZY ZGODNOŚCI LICENCJI

1. W tym przypadku musisz przestrzegać warunków GPLv2 przy łączeniu kodu. Nie możesz skorzystać z warunków określonych w nowszych wersjach GPL.
2. Chociaż możesz opublikować swój projekt (zarówno swoje oryginalne dzieło i / lub dzieło, które otrzymałeś i zmodyfikowałeś) na warunkach GPLv2 lub nowszej wersji, pamiętaj, że drugi kod, którego używasz, musi pozostać na GPLv2. Tak długo, jak twój projekt używa tego kodu, nie będziesz mógł uaktualnić licencji swojego projektu do GPLv3 lub nowszej, a całe dzieło (dowolna kombinacja zarówno twojego projektu, jak i innego kodu) będzie mogło być dostarczane tylko zgodnie z warunkami GPLv2.
3. Jeśli masz możliwość opublikowania projektu na licencji GPLv2 lub dowolnej nowszej wersji, możesz wydać go na licencji GPLv3 lub dowolnej nowszej wersji - a gdy to zrobisz, będziesz mógł włączyć kod wydany na licencji GPLv3.
4. Jeśli masz możliwość publikacji projektu w ramach LGPLv2.1 lub dowolnej nowszej wersji, możesz opublikować go w wersji LGPLv3 lub dowolnej nowszej - a kiedy to zrobisz, będziesz mógł włączyć kod opublikowany w ramach LGPLv3.
5. Musisz postępować zgodnie z warunkami LGPLv2.1, kiedyłączysz kod. Nie możesz skorzystać z warunków określonych w nowszych wersjach GPL.
6. Jeśli to zrobisz, tak długo, jak projekt będzie zawierał kod wydany tylko na LGPLv2.1, nie będziesz mógł uaktualnić licencji do GPLv3 lub nowszej.
7. LGPLv2.1 daje ci możliwość zmiany licencjonowania na dowolną wersję GPL od GPLv2. Jeśli możesz zmienić kod GPL na odpowiednią wersję GPL (jak podano w tabeli), możesz utworzyć tę kombinację.
8. GPLv3 to GPLv3 z dodatkowymi uprawnieniami, które możesz zignorować w tym przypadku.
9. Ponieważ GPLv2 nie zezwala na połączenie z GPLv3, musisz dostarczyć projekt pod warunkami GPLv3 w tym przypadku, ponieważ pozwoli to na połączenie.



ŚWIADOME UŻYWANIE BIBLIOTEK JEST PODSTAWĄ

LISTA BIBLIOTEK KOMPATYBILNYCH Z GPL V2:

Poniżej lista licencji kompatybilnych z GPL v2. Należy pamiętać, że ta kompatybilność nie działa w obie strony, lecz mowa jest tutaj tylko o kompatybilności względem licencji GPL v2.

- Artistic License 2.0 (#ArtisticLicense2)
- Boost Software License
- CeCILL version 2
- Clarified Artistic License
- Cryptix General License
- Eiffel Forum License, version 2
- EU DataGrid Software License
- Intel Open Source License (#intel)
- LGPL – Lesser General Public License, wersji 2.1 (#LGPLv2.1)
- Licencja eCos, w wersji 2.0 (#eCos2.0)
- Licencja Expat (#Expat)
- Licencja FreeBSD (#FreeBSD)
- Licencja FreeType (#freetype)

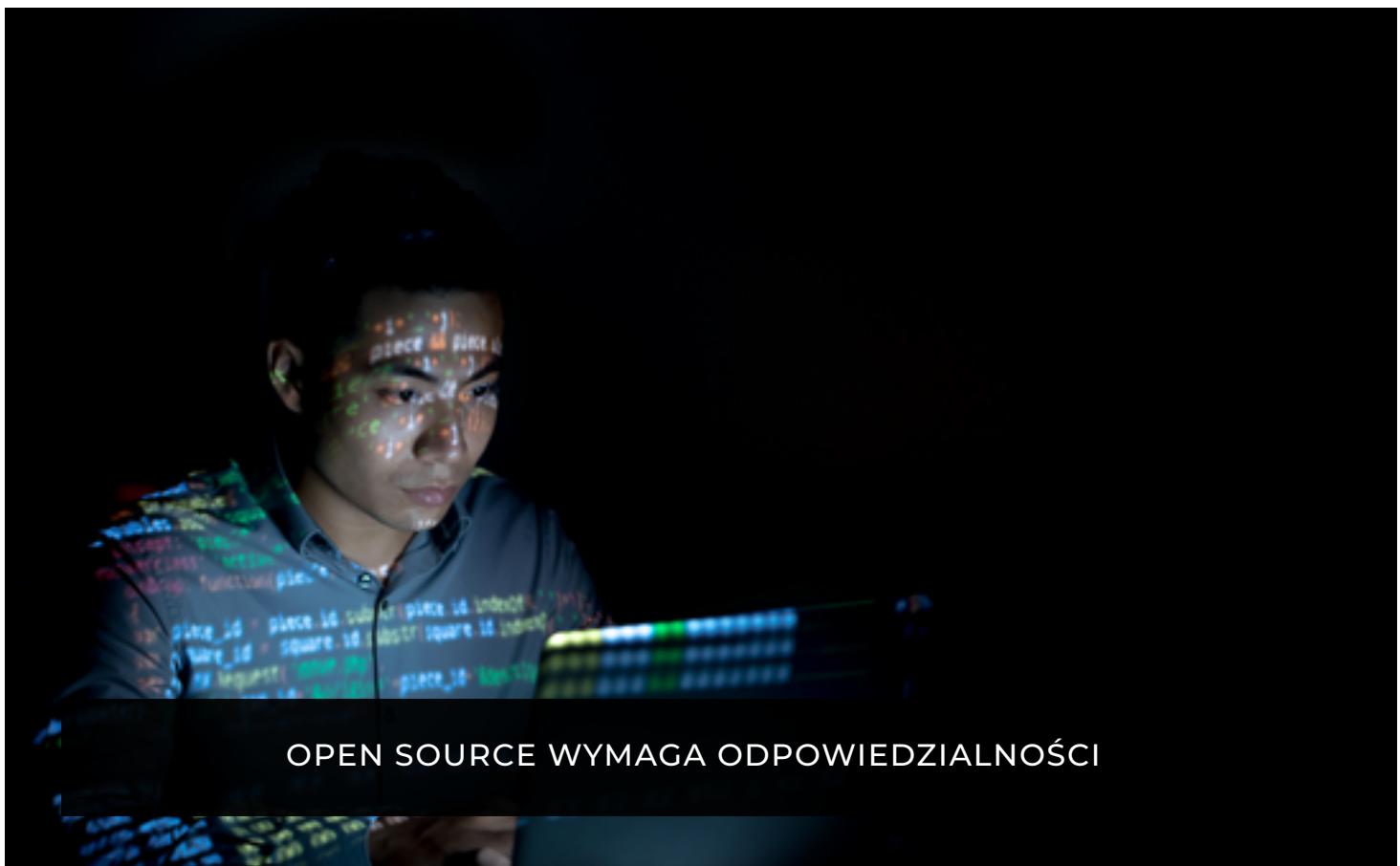
- License of the iMatix Standard Function Library
- Licencja Javascriptu firmy Netscape (#NetscapeJavaScript)
- License of Perl w wersji 5 i wcześniejszych (#PerlLicense)
- License of Python 1.6a2 and earlier versions
- License of Python 2.0.1, 2.1.1, and newer versions
- Licencja języka Ruby (#Ruby)
- License of Vim, version 6.1 or later
- License of ZLib
- BSD license without the advertising clause
- Mozilla Public License (MPL) version 2.0 [The more recent MPL version 2.0 is compatible with the GPL, due to a lot of hard license analysis work... making the MPL compatible with the GPL was one of the main reasons the MPL was updated. However, MPL version 2.0 has optional "incompatible With secondary licenses" clause, and if you include it the software is no longer GPL-compatible (so do not include that clause)].
- OpenLDAP License, version 2.7
- Public Domain
- Sleepycat Software Product License (#BerkeleyDB)
- Standard ML of New Jersey Copyright License
- W3C Software Notice and License
- X11 License / MIT
- Zope Public License, version 2.0
- License of Guile
- License of the run-time units of the GNU Ada compiler

Źródło: http://gplv3.fsf.org/wiki/index.php/Compatible_licenses#GPLv3-compatible_licenses

LISTA BIBLIOTEK KOMPATYBILNYCH Z GPL V3:

Poniżej lista licencji kompatybilnych z GPL v3, należy pamiętać, że ta kompatybilność nie działa w obie strony, lecz mowa jest tutaj tylko o kompatybilności względem licencji GPL v3.

- Apache License, Version 2
- Afferro General Public License, Version 3 (see sec. 13 of GPLv3)
- Lesser General Public License, Versions 2, 2.1 and 3 (LGPL)
- + licencje z GPL v2.



OPEN SOURCE WYMAGA ODPOWIEDZIALNOŚCI

LISTA BIBLIOTEK NIEKOMPATYBILNYCH Z GPL V2:

- Academic Free License, version 1.1.
- Academic Free License, version 2.1.
- Affero General Public License, version 1.0, 2.0
- Aladdin Free Public License (#Aladdin)
- Apache License, version 1.0, 1.1, 2.0
- Apple Public Source License, version 1.0, 1.1, 1.2, 2.0
- Artistic License 1.0 (#ArtisticLicense)
- Common Development and Distribution License
- Common Public License, version 1.0
- Creative Commons
- Eclipse Public License, version 1.0
- GNU Free Documentation License, version 1.2
- GPL for Computer Programs of the Public Administration (#GPL-PA)
- Hacktivismo Enhanced-Source Software License Agreement (HESSLA) (#HESSLA)
- IBM Public License, version 1.0
- Interbase Public License, version 1.0
- Jabber Open Source License, version 1.0

- Jahia Community Source License (#Jahia)
- LaTeX Project Public License 1.2, 1.3a
- Licencja eCos, w wersji 1.1 (#eCos1.1)
- Licencja Lha (#Lha)
- Licencja Shared Source firmy Microsoft, wykorzystana dla CLI, C# i Jscriptu (#Ms-SS)
- Licencja programu PINE (#PINE)
- License of Python 1.6b1 and later versions, through 2.0 and 2.1
- Licencja Publiczna Uniwersytetu Utah (#UtahPublicLicense)
- Licencja programu Scilab (#Scilab)
- License of xinetd
- Licencja YaST (#YaST)
- Lucent Public License version 1.02 (Plan 9 license)
- Mozilla Public License version 1.0, 1.1
- NASA Open Source Agreement
- Netizen Open Source License, version 1.0
- Netscape Public License
- Nokia Open Source License
- Old OpenLDAP License, version 2.3
- Open Public License (#OpenPublicL)
- Open Software License, version 1.0
- OpenSSL license
- Original BSD license
- PHP license, version 3.01
- Phorum License, version 2.0
- Q Public License, version 1.0
- Reciprocal Public License (#RPL)
- Stara licencja ksh93 (#ksh93)
- Stara licencja Plan 9 (#Plan9)
- Stara licencja Squeak (#Squeak)
- Sun Industry Standards Source License 1.0
- Sun Public License
- Sun Community Source License (#SunCommunitySourceLicense)
- SunSolarisSourceCode(Foundation Release) License, Version 1.1 (#SunSolarisSourceCode)
- The Condor Public License
- XFree86 1.1 License
- Zend License, version 2.0
- Zope Public License, version 1

Źródło: http://gplv3.fsf.org/wiki/index.php/Compatible_licenses#GPLv3-compatible_licenses

NAJCZĘŚCIEJ POPEŁNIANE BŁĘDY

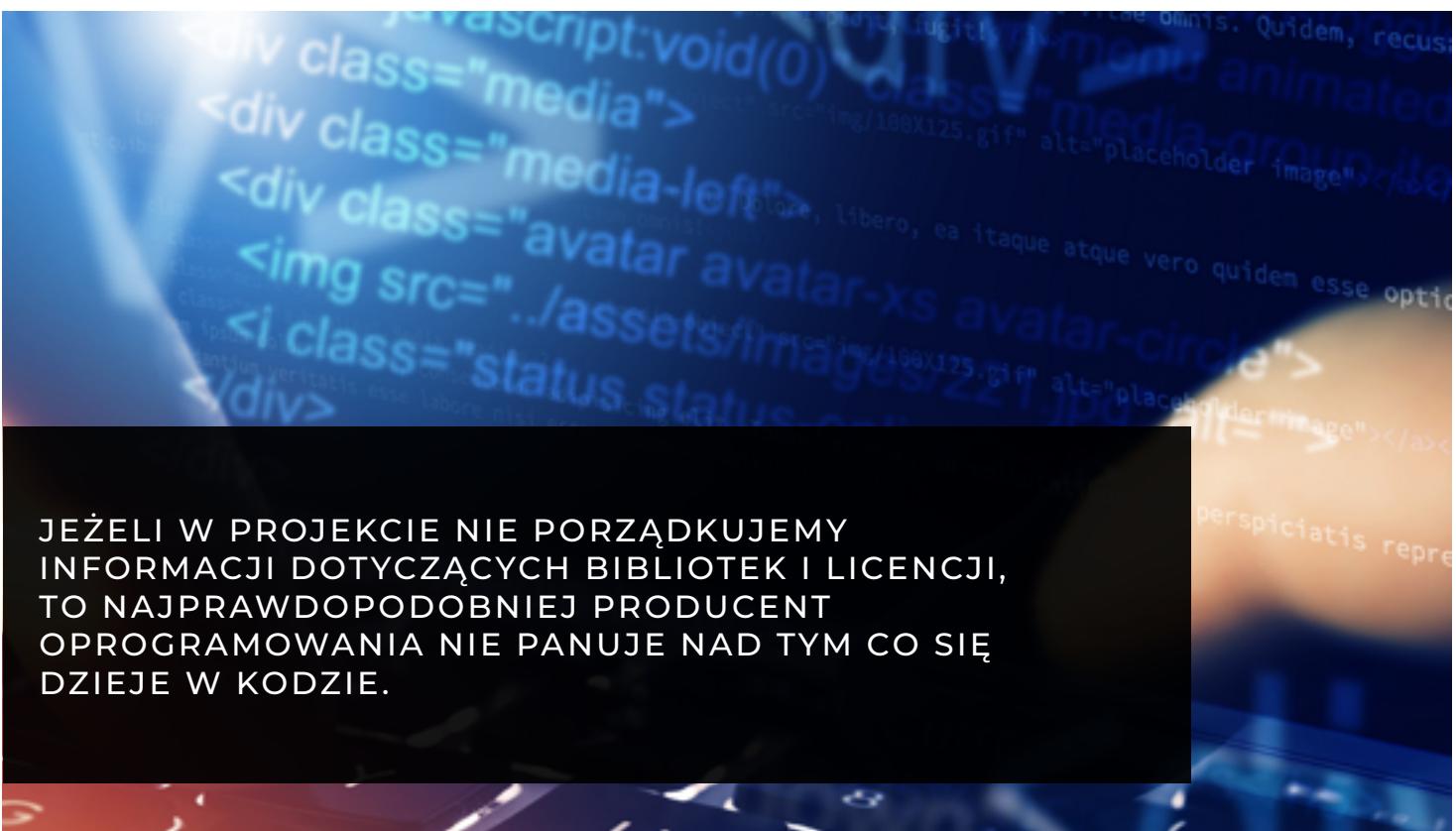
JAKO PRODUCENT OPROGRAMOWANIA BARDZO CZĘSTO ANALIZUJEMY NIE TYLKO KOD INNYCH SYSTEMÓW, ALE RÓWNIEŻ ZAPISY LICENCYJNE, ABY MIEĆ PEWNOŚĆ, CZY JAKĄŚ CZĘŚĆ KODU MOŻEMY WYKORZYSTAĆ WE WŁASNYM ROZWIĄZANIU.

BAŁAGAN W BIBLIOTEKACH I LICENCJACH

Wyszukiwanie licencji i bibliotek w projektach open source z założenia powinno być bardzo proste, ponieważ dobrze napisane oprogramowanie PHP/JS korzysta z:

- composera [<https://getcomposer.org/>] - uporządkowanego zbioru bibliotek PHP
- yarn [<https://yarnpkg.com/lang/en/>] - uporządkowanego zbioru bibliotek JavaScript
- npm [<https://www.npmjs.com/>] - menedżera plików

Na podstawie tych danych można w prosty sposób stworzyć listę bibliotek, zależności i licencji używanych w systemie.



Sprawdziliśmy kilka otwartych systemów klasy CRM:

- **SuiteCRM**

- composer: tylko 10% z listy wszystkich wykorzystywanych bibliotek
- yarn: brak
- npm: brak

- **VtigerCRM**

- composer: brak
- yarn: brak
- npm: brak

- **EPESI**

- composer: tylko 10% z listy wszystkich wykorzystywanych bibliotek
- yarn: brak
- npm: brak

- **JoForce**

- composer: tylko 3% z listy wszystkich wykorzystywanych bibliotek
- yarn: brak
- npm: tylko jedna biblioteka

- **coreBOS**

- composer: tylko jedna biblioteka
- yarn: brak
- npm: brak

Należy sobie zadać pytanie, w jaki sposób można rzeczywiście kontrolować zmiany w systemie, w licencjach i w bibliotekach, skoro nie jest to uporządkowane globalnie dla aplikacji.

COPYLEFT

WYKORZYSTUJĄC BIBLIOTEKI, POZNAJ ZAŁOŻENIA ICH LICENCJI

ROZPOWSZECHNIANIE LICENCJI Z BIBLIOTEKĄ COPYLEFT

LICENCJE COPYLEFT MAJĄ NAJCZĘŚCIEJ TO DO SIEBIE, ŻE JEŻELI DO SWOJEGO OPROGRAMOWANIA WYKORZYSTASZ BIBLIOTEKĘ O LICENCJI WZAJEMNEJ, WÓWCZAS CAŁY SYSTEM MUSI RÓWNIEŻ BYĆ NA TEJ SAMEJ LICENCJI.

EPESI

Przykład takiego problemu można zauważać między innymi w systemie EPESI, który domyślnie rozpowszechnia swoje oprogramowanie na licencji MIT, a w praktyce zawiera biblioteki, które nie są licencyjnie kompatybilne. Do takich bibliotek zalicza się Roundcube, który jest [klientem poczty](#). Dodatkowo można znaleźć inne wirusowe licencje, np. [bibliotekę UIUIKit](#), która jest na licencji AGPL.

Należy zatem zadać pytanie producentowi EPESI, na jakiej licencji jest system MIT? GPLv3? AGPLv3?

joForce

Tutaj występuje problem podobny do sytuacji w systemie EPESI, lecz na większą skalę. Domyślnie system jest na licencji "JoForce Public License 1.0", choć tak naprawdę jest to ta sama licencja co VPL 1.1. Nie wchodząc w szczegóły, na pewno nie jest to licencja GPL/AGPL. To, co rzuca się na pierwszy "rzut oka":

1. Biblioteki na licencji GPLv3:

- Roundcube:

[28](https://github.com/joforce/JoForceCRM/tree/e40d7df2f5774c0c4fb1d380d8eb20407dd55145/modules>EmailPlus/roundcube</p></div><div data-bbox=)

- jbPivot:
https://github.com/joforce/JoForceCRM/blob/270dbecb208904af73e505d65974253b33de1e30/libraries/jquery/pivot/js/formatter_default.js
- aggregate_distinct:
https://github.com/joforce/JoForceCRM/blob/270dbecb208904af73e505d65974253b33de1e30/libraries/jquery/pivot/js/agregate_distinct.js

2. Biblioteki na licencji GPLv2:

- mPDF:
<https://github.com/joforce/JoForceCRM/blob/80cb61182f95174e96830465dab956a80f772369/modules/PDFMaker/mpdf/LICENSE.txt>

3. Biblioteki na licencji AGPLv3:

- ODFViewerPlugin:
<https://github.com/joforce/JoForceCRM/blob/270dbecb208904af73e505d65974253b33de1e30/libraries/jquery/Viewer.js/PluginLoader.js>
- webodf:
<https://github.com/joforce/JoForceCRM/blob/master/libraries/jquery/Viewer.js/webodf.js>
VtigerCRM

VtigerCRM

1. Biblioteki na licencji GPLv3:

- jbPivot:
<http://code.vtiger.com/vtiger/vtigercrm/blob/master/libraries/jquery/pivot/js/jbPivot.min.js>
- freefont:
<http://code.vtiger.com/vtiger/vtigercrm/blob/master/libraries/tcpdf/fonts/freefont/CREDITS>

2. Biblioteki na licencji AGPLv3:

- ODFViewerPlugin:
<http://code.vtiger.com/vtiger/vtigercrm/blob/master/libraries/jquery/Viewer.js/ODFViewerPlugin.js>
- webodf:
<http://code.vtiger.com/vtiger/vtigercrm/blob/master/libraries/jquery/Viewer.js/webodf.js>

Podsumowanie

Konflikty te wskazują, jak wiele jeszcze muszą zmienić u siebie producenci, aby nie łamać licencji zawartych w ich oprogramowaniu. Na szczęście aktualnie istnieje wiele dobrych narzędzi [np.: <https://fossa.io/>], które pozwolą na weryfikację tego, co opisał nam dostawca rozwiązania.

Aby rozwiązać problemy licencyjne, należy zmienić podejście do tworzenia oprogramowania, tak aby kwestie licencyjne zawsze były pilnowane i audytowane na każdym etapie wytwarzania oprogramowania.

ŁĄCZENIE BIBLIOTEK Z LICENCJĄ COPYLEFT

W PUNKCIE POWYŻEJ OMAWIAŁISMY SYTUACJĘ, W KTÓREJ DOŁĄCZAMY BIBLIOTEKĘ COPYLEFT DO NASZEGO SYSTEMU. NATOMIAST W TYM PUNKCIE OMÓWIMY TEMAT BARDZIEJ ZŁOŻONY, CZYLI DOŁĄCZANIE BIBLIOTEK DO SYSTEMU, KTÓRY JEST NA LICENCJI COPYLEFT. SYSTEMÓW, KTÓRE SĄ NA TAKICH LICENCJACH, JEST WIELE, M.IN:

- ▶ OroCRM [OSLv3]
- ▶ SuiteCRM [AGPLv3]
- ▶ Personal CRM [GPLv3]
- ▶ ERPNext [GPLv3]
- ▶ Dolibarr [GPLv3]

W teorii sprawa związana z łączeniem tych bibliotek z innymi bibliotekami jest dość prosta (od strony formalnej), ponieważ w Internecie można znaleźć listę licencji copyleft, które są ze sobą kompatybilne. Dla przykładu w sekcji “Lista bibliotek kompatybilnych z GPL” zostały opisane kompatybilne ze sobą biblioteki.

Spójrzmy, jak taka kompatybilność wygląda w praktyce:

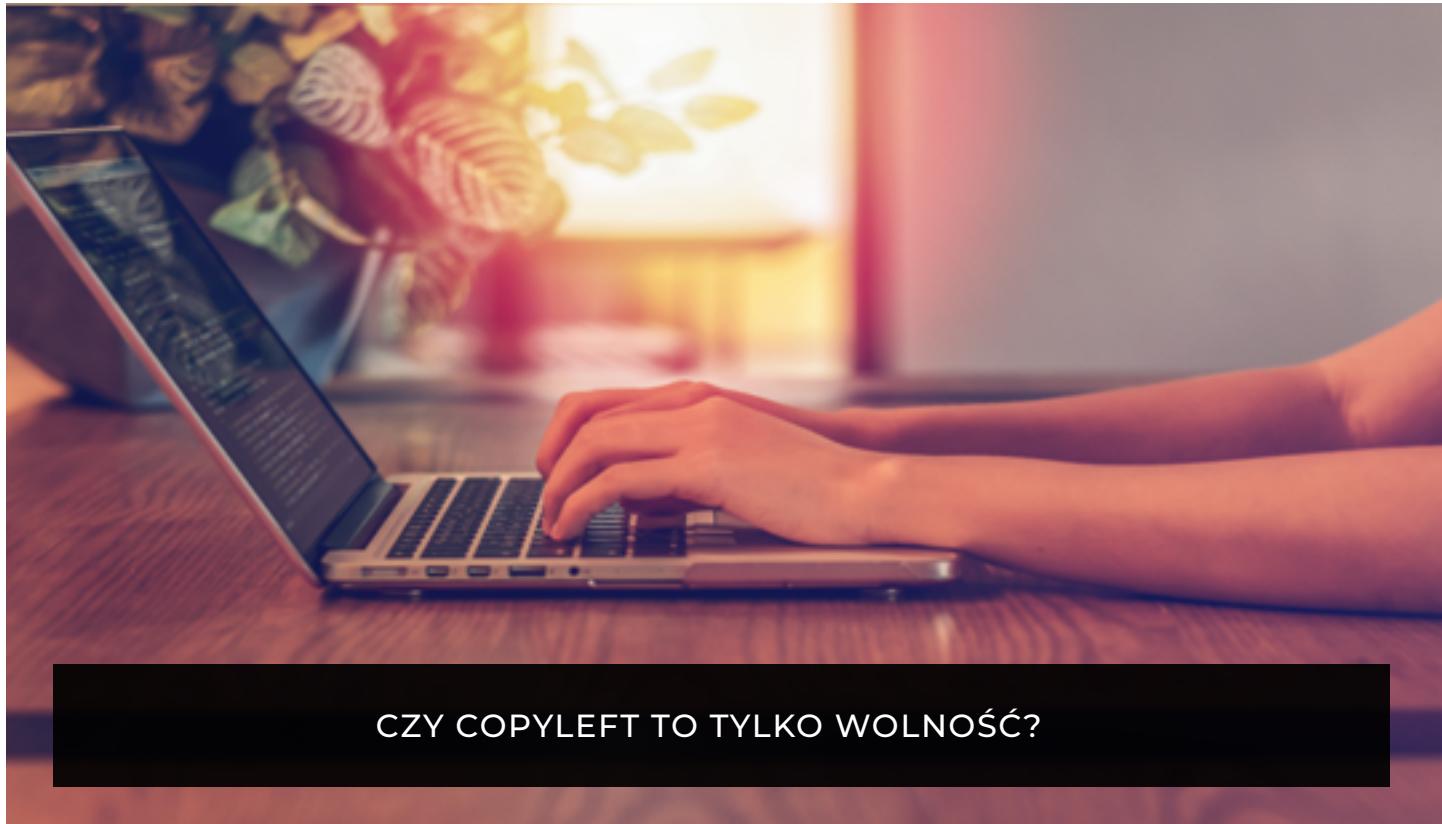
SuiteCRM

SuiteCRM jest na licencji AGPLv3 i można znaleźć w aplikacji następujące biblioteki:

1. rgraph [GPLv2] [niekompatybilna]:
<https://github.com/salesagility/SuiteCRM/blob/master/include/SuiteGraphs/rgraph/license.txt>
2. HTTP_WebDAV_Server [PHPv3] [niekompatybilna]:
https://github.com/salesagility/SuiteCRM/blob/master/include/HTTP_WebDAV_Server/Server.php
3. Crypt_Blowfish [PHPv3] [niekompatybilna]:
https://github.com/salesagility/SuiteCRM/tree/master/include/Pear/Crypt_Blowfish
4. XML_HTMLSax3 [PHPv3] [niekompatybilna]:
https://github.com/salesagility/SuiteCRM/tree/master/include/Pear/XML_HTMLSax3
5. class.soap_transport_http.php [LGPL] [niekompatybilna]:
https://github.com/Seven-7/website-gnuworld/blob/master/php_includes/SOAP/class_nusoap_base.php
6. pclzip [GNU/LGPL] [niekompatybilna]:
<https://github.com/salesagility/SuiteCRM/blob/master/include/pclzip/pclzip.lib.php>

Takich bibliotek w SuiteCRM można znaleźć bardzo dużo. Niestety ze względu na brak jednego centralnego miejsca z listą wszystkich bibliotek ciężko ocenić skalę problemu.

PRZEGŁĄD LICENCJI COPYLEFT



CZY COPYLEFT TO TYLKO WOLNOŚĆ?

OSLV3

“Najnowsze wersje Open Software License zawierają warunek nakazujący dystrybutorom, żeby usiłowali uzyskać od odbiorców wprost wyrażoną akceptację licencji. Co oznacza, że rozprowadzanie oprogramowania OSL przez umieszczenie go na zwykłych serwerach FTP, wysyłanie łat na zwykłe listy wysyłkowe lub przechowywanie tego oprogramowania w zwykłych systemach kontroli wersji jest prawdopodobnie naruszeniem licencji i naraża was na jej odwołanie. Dlatego Open Software License niezwykle utrudnia rozwijanie programów przy użyciu zwykłych narzędzi, z których korzysta się w pracy nad wolnym oprogramowaniem. Z tego powodu, a także dlatego, że jest niezgodna z GPL, zalecamy, żeby nie używać żadnej wersji OSL do żadnego programu.

Stosowanie tej licencji jest bardzo ryzykowne i wymaga głębszej analizy prawnej, aby określić kiedy i od kogo musimy otrzymywać akceptację licencji. Co ciekawe, licencji tej nie zalecają nawet zwolennicy copyleft, ponieważ mocno ogranicza ona rozwój oprogramowania: “Apelujemy, byście nie stosowali Open Software License do pisanej przez siebie oprogramowania. Niemniej jednak, nie ma powodu, by nie korzystać z programów, które zostały wydane na tej licencji”.

AGPLv3

„Jeśli program wykorzystywany jest jedynie wewnętrznie w firmie lub przez osobę prywatną na własny użytek, nie ma konieczności udostępniania publicznie kodu źródłowego takiego oprogramowania. AGPL v.3 wymaga bowiem, by umożliwić użytkownikowi zmodyfikowanego programu pobranie kodu źródłowego, gdy dochodzi do pobrania go „z zewnątrz”. Taka możliwość nie może być jednak wykorzystywana w celu pobierania opłaty za udostępnienie kodu źródłowego programu potencjalnym użytkownikom znajdującym się poza organizacją. W takim przypadku dochodzi bowiem do udostępniania oprogramowania „na zewnątrz”, w związku z czym spełniony zostaje warunek przewidziany w licencji (a zatem zobowiązanie do udostępniania kodu źródłowego za darmo).

To, co może sprawiać problemy z rozróżnieniem pomiędzy udostępnieniem „na zewnątrz” oraz udostępnieniem „wewnętrz” organizacji, nie jest możliwe do jednoznacznego określenia. Z jednej bowiem strony wielotysięczna organizacja posługującą się oprogramowaniem, może nie udostępniać oprogramowania „na zewnątrz”, ale wykorzystywać je na setkach stanowisk swoich pracowników. Z drugiej, pojedyncze osoby przekazujące sobie oprogramowanie mogą w świetle AGPL v.3 być zobligowane do udostępnienia kodu.

Powyższe wynika z faktu, że organizacje (tu: spółki kapitałowe) traktowane są w prawie jako, osobne od pracowników, osoby. Tym samym, jeśli dana organizacja zatrudnia sto osób, to dalej jest jedną organizacją (osobą). Wątpliwości pojawiają się w chwili, w której do serwera „dopuszczane są” osoby formalnie nie będące pracownikami. W Polsce będą to często osoby prowadzące jednoosobową działalność

gospodarczą czy osoby związane umową o dzieło. W przeważającej części przypadków taka sytuacja nie będzie kwalifikowana jako „wewnętrzne” wykorzystanie oprogramowania.”

Wykorzystywanie systemu na licencji AGPLv3 jest dość ryzykowne w organizacji, jeżeli planujemy rozwój i rozbudowę tego systemu, ponieważ każda zmiana musi być udostępniania użytkownikom tego systemu. Niestety większość firm ukrywa tę informację i nie przedstawia swoim klientom opinii prawnej dotyczącej licencji systemu, który jest wdrażany.

„Dopuszczenie do intranetu korporacyjnego lub sieci wewnętrznej osób niebędących pracownikami (jednoosobowi przedsiębiorcy, osoby „na zleceniu”) równoznaczne będzie z rozpowszechnieniem oprogramowania, a tym samym powodować będzie obowiązek udostępnienia kodu. Pamiętać przy tym należy, że każdy przypadek takich działań wymaga dokonania odrębnej oceny oraz kwalifikacji prawnej.

Pamiętać należy, że w praktyce dochodzi do sytuacji o wiele bardziej skomplikowanych. Przykładem może być tworzenie organizacji powiązanych (spółek zależnych), w których jedna będzie „nośnikiem” własności intelektualnej (licencji, znaków itp.), zaś druga „nośnikiem” pracowników. Pierwszy z podmiotów zatrudniać wówczas może pracowników na ułamek etatu jedynie celem nieaktywowania „wirusa copyleft”, zaś podmiotem realnie wypłacającym wynagrodzenie będzie drugi z podmiotów. Tego typu przypadki są jednak zbyt skomplikowane, by analizować je tu „krok po kroku”. W tego typu przypadkach należy kontaktować się z odpowiednim specjalistą”.

GPLv3

Spośród tych trzech licencji copyleft, GPLv3 jest najbardziej przyjazna biznesowo, bo o ile musimy wszystkie zmiany wprowadzać jako GPLv3, o tyle moment, w którym jesteśmy zmuszeni komuś dać kod źródłowy, jest bardziej przyjazny. Przykładowo, gdy zmodyfikujemy system i będziemy go dalej rozpowszechniać, np. poprzez stronę internetową.

Licencje AGPLv3 i OSLv3 są bardzo niekorzystne biznesowo i zalecamy nie wykorzystywać oprogramowania na tych licencjach bez wcześniejszej analizy prawnej.

PUŁAPKI LICENCYJNE

AGPL

Google całkowicie zakazuje używania oprogramowania na licencji AGPL w całej organizacji niezależnie od celu wykorzystania oprogramowania. Dotyczy to również zwykłych aplikacji pobieranych na pulpit naszego komputera.

This page is part of Google's [open source documentation](#).

AGPL Policy

WARNING: Code licensed under the GNU Affero General Public License ([AGPL](#)) MAY NOT be used at Google.

The license places restrictions on software used over a network which are extremely difficult for Google to comply with. Using AGPL software requires that anything it links to must also be licensed under the AGPL. Even if you think you aren't linking to anything important, it still presents a huge risk to Google because of how integrated much of our code is. **The risks heavily outweigh the benefits.**

- Do not attempt to check AGPL-licensed code into google3 or use it in a Google product in any way.
- Do not install AGPL-licensed programs on your workstation, Google-issued laptop, or Google-issued phone without explicit authorization from the Open Source Programs Office.

In some cases, we may have alternative licenses available for AGPL licensed code.

Except as otherwise noted, the content of this page is licensed under [CC-BY-4.0](#) license. Third-party product names and logos may be the trademarks of their respective owners.

SISSL

Kod wydany na podstawie licencji Sun Industry Standards Source License (SISSL) nie powinien być używany w organizacji. Ta licencja ma warunki, które są bardzo trudne do spełnienia (nawet Sun, przed nabyciem, przestał używać lub rekommendować tej licencję).

WTFPL

Kod wydany na podstawie WTFPL nie powinien być używany w organizacjach. Ta licencja ma wiele wad prawnych (brak zrzeczenia się gwarancji, bardzo niejasne przyznanie praw).

SuiteCRM

Wyjątkowym apetytem na ryzyko wykazała się firma stojąca za SuiteCRM, ponieważ postanowiła modyfikować oryginalne zapisy licencji, jakie zostały opublikowane w kodzie przez SugarCRM. Praktycznie wszystkie pliki licencji zostały zmodyfikowane m.in. o:



O ile SuiteCRM ma prawo chronić swoją własność dla swoich modyfikacji, o tyle nie wolno mu ingerować w oryginalną licencję, w szczególności jeżeli w pliku nie dokonano żadnych zmian.

"Licencja AGPL v.3 nie umożliwia modyfikacji warunków tejże licencji. Jeśli dojdzie do takiej modyfikacji, zastosowanie znajdzie tzw. atrybucja, czyli obowiązywać będą podstawowe zapisy licencji, na podstawie której doszło do rozpowszechnienia programu podstawowego.

W praktyce zdarzają się jednak przypadki takich modyfikacji. Pamiętać przy tym należy, że zmiana warunków licencji może wiązać się z istotnym ryzykiem. Pierwotny licencjonodawca może bowiem dochodzić swoich praw na drodze postępowania sądowego od wszystkich podmiotów naruszających jego prawa, nawet jeśli robią to nieświadomie".

MATERIAŁY ZEWNĘTRZNE:

GPLv3 Final Discussion Draft Rationale:
<http://gplv3.fsf.org/gpl3-dd4-rationale.pdf>

Często zadawane pytania na temat GNU GPL:
<https://www.gnu.org/licenses/gpl-faq.html#VersionTwoOrLater>

Przykładowa lista bibliotek zewnętrznych dla SugarCRM 7.9.0.0.:
<https://support.sugarcrm.com/files/d38fd368-35dd-11e7-93de-06ac51574995.pdf>

Macierz zgodności licencji:
<https://www.gnu.org/licenses/gpl-faq.en.html#AllCompatibility>

Make Your Open Source Software GPL-Compatible:
<https://www.dwheeler.com/essays/gpl-compatible.html>

Opinia prawnia dotycząca AGPL:
https://public.yetiforce.com/pdf/opinia_prawna_dla_licencji_agpl.pdf

Niezależna opinia:
<https://opensource.stackexchange.com/questions/5637/use-lGPL-code-in-an-AGPL-project>

Różnica pomiędzy GPL v2 i GPL v3:
<http://www.ifross.org/en/what-difference-between-gplv2-and-gplv3>

Torvalds: No GPL 3 for Linux:
<https://www.cnet.com/news/torvalds-no-gpl-3-for-linux/>

The Lesser GPL:
<https://copyleft.org/guide/comprehensive-gpl-guidech11.html>

Licencje otwartego oprogramowania – praktyczny przewodnik:
<http://lookreatywni.pl/kategorie/prawo-autorskie/licencje-otwartego-oprogramowania/>

Przykładowy opis sprawy o naruszenie zapisów licencji open source:
<https://www.fsf.org/blogs/licensing/update-on-artifex-v-hancom-gnu-gpl-compliance-case-1>

Opis firmy Google dotyczący licencji open source:
<https://opensource.google.com/docs/releasing/>

10 najczęstszych pytań i odpowiedzień dotyczących licencjonowania GPL:
<https://resources.whitesourcesoftware.com/blog-whitesource/top-10-gpl-license-questions-answered>



YETIFORCE

CHOOSE OPEN SOURCE

KONTAKT

TELEFON

+ 48 884 999 123

STRONA WWW

www.yetiforce.com

www.yetiforce.shop

E-MAIL

info@yetiforce.com

YETIFORCE jest producentem jednego z najlepszych systemów CRM na świecie. System jest oparty o otwarty kod źródłowy, a dzięki temu jest nieustannie rozwijany przez jego użytkowników. Dzięki specjalnej, wyjątkowo liberalnej licencji, system jest przyjazny biznesowi. Może być bez przeszkód rozwijany i modyfikowany zgodnie z potrzebami firmy. Jako jeden z najbezpieczniejszych systemów CRM YetiForce podoła nawet najbardziej wymagającym integracjom z systemami otwartymi i komercyjnymi.