

РОЛЬ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ С ОТКРЫТЫМ ИСХОДНЫМ КОДОМ В СОВРЕМЕННОЙ РАЗРАБОТКЕ КОРПОРАТИВНЫХ ИНФОРМАЦИОННЫХ СИСТЕМ

А.А. Шинкарев

ООО «Софтмаст-ИТ», г. Челябинск, Россия

На текущий момент существует множество программных продуктов или пакетов с открытым исходным кодом, и их количество с каждым днем увеличивается. Из этого можно сделать вывод о том, что публикация исходного кода становится все более и более популярным веянием в мире разработки программного обеспечения. Особое внимание при публикации исходного кода программного решения или пакета для его использования в сообществе разработчиков следует уделить типу лицензии – от этого зависит то, какие сценарии будут доступны для использования с применением опубликованного пакета или программного решения. Также необходимо составить полную и подробную документацию и определиться со способами продвижения публикуемого пакета в среде разработчиков. **Цель исследования.** Обосновать целесообразность и необходимость публикации программных продуктов, пакетов и библиотек для использования их другими разработчиками для построения собственных систем и сервисов. Привести описание основных типов лицензий открытого программного обеспечения, выявить их особенности и отличия, а также описать те ситуации, для которых тот или иной тип лицензии подходит в большей или меньшей степени. Обосновать необходимость написания документации. Описать способы продвижения и популяризации в сообществе разработчиков публикуемых программных продуктов, пакетов и библиотек. **Материалы и методы.** Рассматриваются официальные документы лицензий с описанием условий использования, воспроизведения и распространения. Анализируются основные пути и способы продвижения программных продуктов с открытым исходным кодом. **Результаты.** В статье автором обосновывается актуальность публикации и использования исходного кода разработанного программного продукта, пакета или библиотеки. Описываются основные положения наиболее распространенных типов лицензий. Также приводятся советы по выбору типа лицензии при публикации исходного кода для свободного использования. Обосновывается необходимость написания документации по публикуемому программному продукту. Описываются некоторые способы для продвижения опубликованных пакетов, например, такие как грамотный выбор имени, выступление на конференциях или публикация статей с примерами использования.

Ключевые слова: корпоративные информационные системы, программное обеспечение с открытым исходным кодом, MIT, Mozilla Public License 2.0, Apache License 2.0.

Введение

Сегодня все большее значение приобретает открытость компаний, занимающихся разработкой программного обеспечения. Под открытостью подразумевается не столько финансовая прозрачность или контролируемость внутрикорпоративных процессов извне, сколько их технологическая открытость, их приверженность сообществу свободного программного обеспечения (Free Software Community) [1]. Что же такое технологическая открытость ИТ-компаний здесь и сейчас? Под технологической открытостью с одной стороны можно понимать степень вовлеченности ресурсов компании в разработку программного обеспечения с открытым исходным кодом, а с другой – стремление повторно использовать решения, разработанные другими членами сообщества.

Практически ни один современный проект по разработке программного обеспечения не обходится без использования открытых и в большинстве своем бесплатных программных пакетов, ускоряющих разработку и снижающих ее конечную стоимость. Однако пока не так много компаний рассматривают возможность публикации своих наработок в открытом доступе. Помимо нежелания участвовать в разработке программного обеспечения с открытым исходным кодом многие также не считают целесообразным поддерживать существующие пакеты и вносить в них изменения для своих нужд. Вместо этого начинается разработка своего решения уже в каком-то

виде решенной задачи, что в некотором смысле приводит к освоению бюджета на разработку и созданию интересных задач для программистов вместо решения проблем, стоящих перед бизнес-пользователями приложений.

Критически важным навыком для ИТ-подразделения современных компаний становится умение соблюдать баланс между разработкой закрытой кодовой базы приложений и участием в разработке программного обеспечения с открытым исходным кодом для тех частей приложения, которые не несут бизнес-ценности, а лишь сопровождают выполнение основных целей. Типичным примером внутренней разработки, которую нужно либо выносить в открытое пространство, либо полностью заменять на использование готового, является разработка своих библиотек компонентов пользовательского интерфейса, а также создание своих инструментов поддержки и сопровождения активностей по развертыванию приложений (DevOps) [2], подсистем аутентификации и авторизации, шаблонов веб-приложений и серверных сервисов, то есть любых программных решений, которые не несут уникальной для бизнеса ценности.

1. Основные типы лицензий программного обеспечения с открытым исходным кодом

На сегодняшний день существует довольно внушительное количество лицензий для программного обеспечения с открытым исходным кодом. Ознакомиться с достаточно полным списком широко распространенных лицензий для программного обеспечения можно здесь [3].

Предлагается рассмотреть лишь несколько типов лицензий, которые позволяют публиковать исходный код программных решений с наиболее гибкими условиями использования и распространения и в то же время налагают минимальное количество ограничений, что упрощает использование программного кода в коммерческой корпоративной разработке.

Как и в случае с началом работы с новым программным пакетом, когда важна простота и минимальное время до получения первых результатов работы, в случае с лицензированием важно то, насколько просто работать с лицензией и насколько понятны и прозрачны ее условия.

При описании особенностей лицензий использовались следующие ресурсы [4–8].

1.1. Лицензия MIT

Представляет собой разрешительную лицензию, разрешает делать с кодом практически все, что угодно, при условии, что сама лицензия и указание авторства останутся без изменений даже в случае значительной реструктуризации первоначального программного обеспечения, поставляемого под этой лицензией.

У этой лицензии есть один существенный минус – она не гарантирует пользователю патентных прав [9].

Данная лицензия может хорошо подойти для ситуаций, когда компания хочет поделиться техническими наработками, не дающими бизнесу конкурентных преимуществ, но позволяющими повторно использовать удачные решения в других проектах.

1.2. Лицензия Apache License 2.0

Еще одна разрешительная лицензия, схожая с MIT, но с несколькими важными отличительными чертами. Во-первых, при использовании программного обеспечения, выпущенного под данной лицензией, обязательно указание изменений, внесенных в оригинальное решение. Во-вторых, для производных решений нельзя использовать те же названия, если они являются торговыми.

Важным преимуществом по сравнению с MIT является предоставление патентных прав на производные программные продукты.

Данная лицензия подходит в случае, когда необходимо опубликовать в открытом доступе решение, представляющее собой продукт с определенным названием, которое используется при его распространении. Продукт может быть как полностью бесплатным, так и комбинировать платное и бесплатное распространение в разных сценариях использования. Например, эта лицензия подойдет для публикации исходного кода мобильного приложения, что делает логику его работы прозрачной, но и защищает уже выбранное имя от нежелательного повторного использования. Другим примером может быть исходный код облачного сервиса, который выполняет определенную ценную работу и предоставляется как платно по подписке, так и бесплатно в случае самостоятельного развертывания и поддержки на внутрикорпоративных серверных мощностях.

1.3. Mozilla Public License 2.0

Данная лицензия хорошо подходит для разработки библиотек из-за слабых правил копиленфта, когда нет необходимости распространять исходный код библиотеки вместе с программным обеспечением, в котором она используется. Однако эта лицензия предполагает лицензирование производных программных продуктов под той же лицензией, что важно в том случае, когда нежелательно, чтобы доработки библиотеки имели, например, более ограничивающую лицензию, чем задумывалось автором.

Также в производных работах защищены названия, используемые в проекте, если они являются торговыми марками.

2. Сценарии создания программного обеспечения с открытым исходным кодом в рамках внутрикорпоративной разработки

В современных условиях создания внутрикорпоративного программного обеспечения можно выделить три основных направления.

1. Разработка клиентских веб-приложений или же фронтенд-разработка (frontend development).
2. Разработка серверной части информационных решений или бекенд-разработка (backend development).
3. Работа в направлении автоматизации развертывания и управления жизненным циклом информационных систем или девопс (DevOps).

В рамках каждого направления разработки сегодня возможно и целесообразно ведение активности по публикации наработок в открытое пространство для повторного использования, в том числе другими компаниями. Использование публичных репозиторий (repository) с исходным кодом библиотек в сочетании с публикацией их сборок в публичные регистры, такие как npm, pypi [10], ведут не только к большей прозрачности и надежности, но и создают положительный образ компании в глазах разработчиков, что, в свою очередь, упрощает процесс поиска кандидатов на открывающиеся вакансии.

Рассматривая фронтенд-разработку и наиболее распространенные сценарии создания разделяемых пакетов, в том числе и не доступных публично, можно выделить задачу создания библиотеки компонентов пользовательского интерфейса для той или иной платформы, таких как React, Angular или Vue [11].

Само по себе создание подобной библиотеки компонентов пользовательского интерфейса позволяет решить задачу создания общего стиля группы приложений, в которых они используются, получить так называемый стайлгайд (style guide). Однако стоит всегда обращать внимание на существующие библиотеки, ведь для бизнеса может быть куда более практичной доработка существующего решения под себя, нежели написание своего решения с нуля, при котором придется столкнуться со всеми подводными камнями, которые уже во многом были кем-то преодолены ранее.

Также распространенной задачей может стать создание шаблонов приложений, которые позволяют осуществить быстрый старт нового проекта, что актуально как для фронтенд-, так и для бекенд-разработки. В данном случае может быть целесообразно поддерживать такие шаблоны в открытом доступе и актуализировать параллельно с текущей работой над проектами, которые были созданы с их помощью.

В бекенд-разработке актуально создание различных пакетов, скрывающих в себе, например, логику работы с JWT-аутентификацией [12] или контракт получения данных для страничного запроса данных с фильтрацией и сортировкой. Подобные пакеты зачастую нацелены не только на язык программирования, используемый конкретной командой, но и на платформу создания сервисов, как например Asp.NET Core, NodeJS, Flask и т. д.

В сфере работ по направлению девопс наиболее распространенной задачей, решаемой всеми компаниями в том или ином виде, является автоматизация процессов сборки и доставки решений на продуктивное окружение (production environment). Такие решения могут хорошо обобщаться и быть применимы и к другим проектам. Зачастую предпосылками для универсализации является использование командами разработки таких практик, как GitFlow [13], Conventional Commit [14], Semantic Versioning [15]. Решения, построенные и автоматизированные с учетом этих подходов,

хорошо обобщаются на другие проекты за счет того, что это уже стало корпоративным стандартом в отрасли разработки программного обеспечения.

3. Продвижение программного обеспечения с открытым исходным кодом

Несмотря на всю увлекательность процесса создания и публикации потенциально полезного для других программного кода, существует еще одна важная задача, идущая рука об руку с созданием библиотеки, а именно – продвижение. Важно не только написать полезный программный модуль, но и суметь рассказать о нем, чтобы те, кому он может пригодиться, смогли найти его на просторах Интернета, где скорее всего уже существуют подобные пакеты. Одним из самых важных атрибутов программного продукта с открытым исходным кодом является его имя. Выбору подходящего имени пакета стоит уделить особое внимание. Необходимо решить, будет ли оно содержать в себе имя компании или будет иметь имя, говорящее о функциональности, или наоборот – не иметь ничего общего с назначением.

Помимо имени пакета также особое внимание нужно уделить документации и времени, которые необходимо, чтобы начать работать с библиотекой. Чем проще процесс быстрого старта, тем больше шансов на то, что, попробовав библиотеку, пользователь продолжит с ней работать и изучать документацию для более сложных и реалистичных случаев использования. Чем больше времени нужно, чтобы начать работать с программным пакетом, и чем хуже составлена документация, тем меньше шансов, что пакетом будет пользоваться кто-то, кроме авторов.

Дополнительно к документации пакета полезным является написание статей, которые бы рассказывали о библиотеке, о том, какие задачи она решает, приводили бы примеры использования. При написании статей имеет смысл учитывать ту информацию, которую разработчики ищут в Интернете для решения схожих задач, анализируя поисковые запросы как при SEO-оптимизации содержимого сайтов. Используя данные о поисковых запросах, необходимо в том числе оптимизировать текст статей, чтобы они попадали в поисковую выдачу разработчиков.

Выступления на конференциях, где можно продемонстрировать разработанную библиотеку, также положительно влияют на продвижение разработанного программного обеспечения в обществе разработчиков.

Заключение

Современный мир разработки программного обеспечения все сильнее ориентируется на модные тенденции. И одной из таких тенденций является разработка программного обеспечения с открытым исходным кодом. Компаниям, которые смогут органично вписаться в этот тренд, будет проще привлекать новых талантливых сотрудников. Ведь то, в каких публичных проектах участвует компания и насколько они полезны и известны, напрямую влияет на имидж компании на рынке труда.

Помимо репутационной составляющей не менее важным фактором, побуждающим участвовать в разработке программного обеспечения с открытым исходным кодом и публиковать уже существующие наработки, является решение задачи поддержки и развития проектов не только своими, зачастую ограниченными силами, но и привлекая разработчиков на просторах Интернета. Поразительно, насколько сильно на качество программного кода влияет осознание того факта, что любой сможет его увидеть и оценить. Эта открытость заставляет большинство участников подобных проектов внимательнее относиться к написанию программного кода, комментарию к нему, документированию функционала, тестированию и даже стилю написания текста коммитов (commits).

Когда компания подходит к этапу своего развития, на котором ей есть чем поделиться с обществом, встает выбор типа лицензии для программного обеспечения с открытым исходным кодом. В зависимости от нужд пакета и наличия или отсутствия в нем торговых марок подойдет одна из рассматриваемых в статье лицензий. Если планируется создание библиотеки общего назначения, то хорошо подойдет лицензия MIT. Если речь идет о выпуске на рынок продукта с открытым исходным кодом, то имеет смысл выбирать между рассматриваемыми Apache License 2.0 и Mozilla Public License 2.0. Безусловно, существует намного больше типов лицензий, но на начальных этапах может быть достаточно тех, что рассматриваются в статье. Также не стоит публиковать программный код без указания лицензии, так как это может привести к неожиданным

последствиям из-за нюансов в законодательстве, касающемся авторских и патентных прав в разных странах.

В рамках создания по-настоящему успешного программного обеспечения с открытым исходным кодом необходимо уделять достойное внимание продвижению создаваемого решения среди разработчиков. Зачастую можно наблюдать плачевную ситуацию, когда хорошо проработанный и потенциально полезный для многих пакет так и не находит свою аудиторию по той причине, что о нем практически невозможно узнать и очень сложно различить его достоинства по сравнению с уже существующими решениями в этой нише. Можно даже сформулировать задачу популяризации решения с тем же приоритетом, что имеет разработка самого программного обеспечения вместе с написанием исчерпывающей и хорошо структурированной документации, ведь это напрямую влияет в том числе на мотивацию осуществлять дальнейшую поддержку и развитие разработанного программного обеспечения.

Несмотря на все нюансы и неоднозначное первое впечатление, которое может возникнуть у бизнеса в ответ на идею выносить части программных продуктов в открытый доступ, разработка программного обеспечения с открытым исходным кодом становится все более популярной. Компании, которые смогут встроить участие в разработке такого программного обеспечения в свои бизнес-процессы, будут выделяться среди полностью закрытых конкурентов и смогут не выпасть из обоймы в будущем, когда ведение разработки программного обеспечения частично в открытом пространстве будет стандартной практикой в ИТ-индустрии.

Литература

1. *The Free Software Community After 20 Years: With great but incomplete success, what now?* – <https://www.gnu.org/philosophy/use-free-software.en.html> (дата обращения: 01.04.2021).
2. *Site Reliability Engineering. Надежность и безотказность как в Google* / Б. Бейер, К. Джонс, Д. Петофф, Н. Мерфи. – СПб.: Питер, 2019. – 592 с.
3. Категория: Некоммерческие лицензии. – [http://licenseit.ru/wiki/index.php/Категория: Некоммерческие_лицензии](http://licenseit.ru/wiki/index.php/Категория:Некоммерческие_лицензии) (дата обращения: 01.04.2021).
4. Лицензия для вашего open-source проекта. – <https://habr.com/ru/post/243091/> (дата обращения: 01.04.2021).
5. Сравнение open-source лицензий. – <https://wiki.merionet.ru/servernye-resheniya/54/sravnienie-open-source-licenzij/> (дата обращения: 01.04.2021).
6. Сравнительный анализ основных лицензий open-source: GPL, LGPL, BSD, MIT, Mozilla public license, Apache software license. – <http://libertarium.ru/18586.html> (дата обращения: 01.04.2021).
7. В чем разница между популярными open-source лицензиями? Объясняет Github. – <https://tproger.ru/articles/whats-difference-between-licenses/> (дата обращения: 01.04.2021).
8. Licenses. – <https://choosealicense.com/licenses/> (дата обращения: 01.04.2021).
9. Открытый код и интеллектуальная собственность. – [https://habr.com/ru/company/mirantis_openstack/blog/272405/](https://habr.com/ru/company/mirantis/openstack/blog/272405/) (дата обращения: 01.04.2021).
10. List of software package management systems. – https://en.wikipedia.org/wiki/List_of_software_package_management_systems (дата обращения: 01.04.2021).
11. Angular vs React vs Vue: Which Framework to Choose in 2021. – <https://www.codeinwp.com/blog/angular-vs-vue-vs-react/> (дата обращения: 01.04.2021).
12. Introduction to JSON Web Tokens. – <https://jwt.io/introduction> (дата обращения: 01.04.2021).
13. A successful Git branching model. – <https://nvie.com/posts/a-successful-git-branching-model/> (дата обращения: 01.04.2021).
14. Conventional Commits. – <https://www.conventionalcommits.org/en/v1.0.0/> (дата обращения: 01.04.2021).
15. Semantic Versioning. – <https://semver.org/> (дата обращения: 01.04.2021).

Шинкарев Александр Андреевич, канд. техн. наук, инженер-программист, ООО «Софт-маст-ИТ»; sania.kill@mail.ru.

Поступила в редакцию 13 апреля 2021 г.

ROLE OF OPEN SOURCE SOFTWARE IN MODERN DEVELOPMENT OF ENTERPRISE INFORMATION SYSTEMS

A.A. Shinkarev, sanja.kill@mail.ru

LLC "Softmast-IT", Chelyabinsk, Russian Federation

At the moment there are many open source software products and packages, and their number is increasing every day. So it can be concluded that publishing source code is becoming more and more popular in the world of software development. When publishing the source code of a software solution or software package for use in the developer community, special attention should be given to the license type – this affects which scenarios will be available for use of the published package or software solution. It is also necessary to draw up full and detailed documentation and decide on the ways to promote the published package among developers. **The purpose of the study** was to justify the feasibility and necessity of publishing software products, packages and libraries for their use by other developers to build their own systems and services. The author meant to describe the major open source licenses, identify their features and differences, and those situations for which this or that type of license is suitable, as well as to demonstrate the need of writing documentation and describe ways to promote and popularize published software products, packages, and libraries in the developer community. **Materials and methods.** The paper considers official license documents describing conditions of use, reproduction, and distribution. The author analyzes the main ways and means to promote open source software products. **Results.** The article substantiates the relevance of publishing and using the source code of a software product, package or library. The author describes the main provisions of the most common licenses and gives advice on choosing the type of license when publishing source code for free use. The necessity of writing documentation for the published software product is substantiated. The article also describes some of the ways to promote published packages, such as the choice of name, speaking at conferences, and publishing articles with case studies.

Keywords: enterprise information systems, open source software, MIT, Mozilla Public License 2.0, Apache License 2.0.

References

1. *The Free Software Community After 20 Years: with Great but Incomplete Success, What Now?* Available at: <https://www.gnu.org/philosophy/use-free-software.en.html> (accessed 01.04.2021).
2. Bejer B., Dzhouns K., Petoff D., Merfi N. *Site Reliability Engineering. Nadezhnost' i bezotkaznost' kak v Google* [Site Reliability Engineering. Reliability and Reliability like in Google]. St. Petersburg, Piter Publ., 2019. 592 p.
3. *Kategoriya: Nekommercheskie licenzii* [Category: Uncommercial licenses]. Available at: http://licenseit.ru/wiki/index.php/Категория:Некоммерческие_лицензии (accessed 01.04.2021).
4. *Licenziya dlya vashego open-source proekta* [License for You Open-source Project]. Available at: <https://habr.com/ru/post/243091/> (accessed 01.04.2021).
5. *Sravnienie open-source licenzij* [Comparison of Open-source Licenses]. Available at: <https://wiki.merionet.ru/servernye-resheniya/54/sravnienie-open-source-licenzij/> (accessed 01.04.2021).
6. *Sravnitel'nyj analiz osnovnyh licenzij open-source: GPL, LGPL, BSD, MIT, Mozilla public license, Apache software license* [Comparison Analysis of the Main Open-source Licenses: GPL, LGPL, BSD, MIT, Mozilla Public License, Apache Software License]. Available at: <http://libertarium.ru/18586.html> (accessed 01.04.2021).
7. *V chem raznica mezhdu populyarnymi open-source licenziyami? Ob'yasnyaet Github* [What is the Difference Between the Popular Open-source Licenses? Github Explains]. Available at: <https://tproger.ru/articles/whats-difference-between-licenses/> (accessed 01.04.2021).
8. *Licenses*. Available at: <https://choosealicense.com/licenses/> (accessed 01.04.2021).
9. *Otkrytyj kod i intellektual'naya sobstvennost'* [Open-source and Intellectual Property]. Available at: https://habr.com/ru/company/mirantis/_openstack/blog/272405/ (accessed 01.04.2021).

10. *List of Software Package Management System*. Available at: https://en.wikipedia.org/wiki/List_of_software_package_management_systems/ (accessed 01.04.2021).

11. *Angular vs React vs Vue: Which Framework to Choose in 2021*. Available at: <https://www.codeinwp.com/blog/angular-vs-vue-vs-react/> (accessed 01.04.2021).

12. *Introduction to JSON Web Tokens*. Available at: <https://jwt.io/introduction> (accessed 01.04.2021).

13. *A Successful Git Branching Model*. Available at: <https://nvie.com/posts/a-successful-git-branching-model/> (accessed 01.04.2021).

14. *Conventional Commits*. Available at: <https://www.conventionalcommits.org/en/v1.0.0/> (accessed 01.04.2021).

15. *Semantic Versioning*. Available at: <https://semver.org/> (accessed 01.04.2021).

Received 13 April 2021

ОБРАЗЕЦ ЦИТИРОВАНИЯ

Шинкарев, А.А. Роль программного обеспечения с открытым исходным кодом в современной разработке корпоративных информационных систем / А.А. Шинкарев // Вестник ЮУрГУ. Серия «Компьютерные технологии, управление, радиоэлектроника». – 2021. – Т. 21, № 2. – С. 16–22. DOI: 10.14529/ctcr210202

FOR CITATION

Shinkarev A.A. Role of Open Source Software in Modern Development of Enterprise Information Systems. *Bulletin of the South Ural State University. Ser. Computer Technologies, Automatic Control, Radio Electronics*, 2021, vol. 21, no. 2, pp. 16–22. (in Russ.) DOI: 10.14529/ctcr210202