

# **SaaS, PaaS, IaaS и не только: модели лицензирования программного обеспечения**

В сегодняшнем быстро развивающемся цифровом пространстве программное обеспечение (далее – «ПО») стало неотъемлемой частью бизнеса в различных отраслях экономики. С появлением облачных вычислений и других новых технологий возник ряд механизмов или моделей лицензирования прав на использование ПО, каждая из которых имеет свои плюсы и минусы. Непрофессионалам может быть непросто ориентироваться в данной юридической области и понять, какая модель может быть наиболее подходящей для их нужд. Цель этой статьи состоит в том, чтобы (1) представить шесть основных моделей лицензирования ПО, начиная от традиционной передачи ПО для установки в сеть пользователя и заканчивая лицензированием ПО с открытым исходным кодом и бесплатным ПО, а также (2) описать их основные характеристики, преимущества и недостатки. Независимо от того, являетесь ли вы пользователем ПО или разработчиком, это краткое руководство поможет вам принять более грамотное решение о том, какую модель лицензирования ПО следует использовать для ваших целей.

Мы понимаем, что у каждого читателя есть свой лимит времени на чтение такого рода статей. Мы структурировали эту статью и разбили её на три части таким образом, чтобы предоставить необходимую полезную информацию каждому читателю.

## **Базовые модели лицензирования ПО**

Для занятых профессионалов, которым нужна краткая информация, мы приводим ниже краткий обзор шести основных моделей лицензирования ПО.

### **Традиционная модель лицензирования:**

В этой модели ПО приобретается или лицензируется клиентом и устанавливается в его собственной компьютерной системе. Клиент владеет и управляет оборудованием и инфраструктурой, необходимыми для запуска ПО. ПО может быть настроено или изменено заказчиком в соответствии со своими конкретными потребностями, а обновления или усовершенствования могут быть предоставлены разработчиком.

## **Программное обеспечение как услуга (Software as a Service, или SaaS):**

В этой модели ПО размещается «в облаке» у провайдера, а клиент получает к нему доступ через сеть «Интернет» на основе подписки. Клиент не владеет и не управляет базовой инфраструктурой, которая в данном случае управляется и обслуживается поставщиком (разработчиком). ПО обычно предлагается на основе платы за пользование (pay-per-use) или с оплатой за подписку (pay-per-subscription), при этом поставщик несет ответственность за безопасность (в том числе данных пользователя), производительность и обслуживание ПО. Обновления ПО становятся обычно доступными для клиентов автоматически без необходимости дополнительной установки или настройки ПО.

## **Платформа как услуга (Platform as a Service, или PaaS):**

В этой модели поставщик предлагает платформу, которую клиенты могут использовать для разработки, запуска и управления своими собственными приложениями. Платформа включает в себя необходимую инфраструктуру и инструменты, такие как операционная система, базы данных и среда разработки. Поставщик управляет базовой инфраструктурой, а клиент отвечает за разработку и поддержку своих собственных приложений. Выставление счетов обычно основано на использовании, таком как объем хранилища данных или вычислительное время, использованное клиентом.

## **Инфраструктура как услуга (Infrastructure as a Service, или IaaS):**

В этой модели провайдер предлагает вычислительную инфраструктуру, такую как виртуальные машины, хранилище и сеть. Клиент контролирует операционную систему, приложения и развернутый контент и может использовать инфраструктуру для разработки, запуска и управления своими собственными приложениями. Поставщик отвечает за управление базовой инфраструктурой, включая аппаратные и сетевые компоненты, а клиент

отвечает за настройку и поддержку своих собственных приложений. Выставление счетов обычно основано на использовании, таком как объем хранилища данных или вычислительное время, использованное клиентом.

## **ПО с открытым исходным кодом (Open source software):**

В этой модели ПО доступно по лицензии с открытым исходным кодом, которая позволяет любому лицу модифицировать, использовать и распространять код. Пользователи, как правило, могут свободно модифицировать ПО в соответствии со своими конкретными потребностями, а обновления и улучшения могут быть (или должны быть в зависимости от типа лицензии (см. об этом нашу отдельную [статью](#))) предоставлены всем пользователям. ПО обычно доступно бесплатно, а доход может быть получен за счет поддержки, консультаций или дополнительных услуг.

## **Бесплатное ПО (Freemium software):**

В этой модели базовая версия ПО предлагается бесплатно, а дополнительные функции доступны для разовой покупки или подписки. Бесплатная версия обычно ограничена с точки зрения функциональности или использования, и пользователям может быть предложено перейти на платную версию ПО, чтобы получить доступ к дополнительным функциям или снять ограничения. Доход также может быть получен за счет рекламы или иных средств маркетинга.

## **Кратко о преимуществах и недостатках каждой модели ПО**

Для читателей, у которых есть больше времени – продолжайте читать. В данной части мы более подробно рассмотрим основные плюсы и минусы каждой модели лицензирования ПО. Мы сделаем данную оценку как с точки зрения разработчика, так и с точки зрения пользователя.

### **Плюсы и минусы традиционной модели лицензирования ПО**

- Перспектива разработчика: разработчики имеют полный контроль над ПО, включая дистрибуцию, обновления и лицензирование. Эта модель может обеспечить более высокую норму прибыли для разработчиков.
- Перспектива клиента: клиенты имеют больший контроль над своими данными и возможность использовать ПО даже без подключения к сети «Интернет».

Однако клиенты самостоятельно несут бремя обслуживания ПО и обеспечения его правильной работы.

## **Плюсы и минусы ПО как услуги (SaaS)**

- Перспектива разработчика: разработчикам легче предоставлять обновления и осуществлять обслуживание, если приложение работает на их собственных серверах. Модель на основе подписки обеспечивает стабильный доход для разработчиков.

- Перспектива клиента: клиентам не нужно беспокоиться об обслуживании или обновлении ПО. Оплата за услугу обычно происходит на регулярной основе, которая, правда, с течением времени может вылиться в значительную сумму. У ряда клиентов могут возникать опасения по поводу безопасности и конфиденциальности своих данных.

## **Плюсы и минусы платформы как услуги (PaaS)**

- Перспектива разработчика: клиенты-разработчики могут сосредоточиться на создании приложений, а не на управлении всей инфраструктурой. Провайдер заботится об обслуживании сервера, его масштабируемости и безопасности.

- Перспектива клиента: клиенты могут сократить расходы на IT-инфраструктуру и сосредоточиться на разработке собственных приложений. Однако они могут быть ограничены функциями и услугами, предоставляемыми платформой.

## **Плюсы и минусы инфраструктуры как услуги (IaaS)**

- Перспектива разработчика: клиенты-разработчики имеют полный контроль над программным стеком, включая операционную систему, приложения и сеть. Эта модель обеспечивает гибкость с точки зрения масштабирования и настройки.

- Перспектива клиента: клиентам не нужно беспокоиться об управлении собственным оборудованием и инфраструктурой. Они также могут быстро увеличивать или уменьшать свои ресурсы по мере необходимости. Однако эта модель требует больше технических знаний и может быть дороже.

## **Плюсы и минусы ПО с открытым исходным кодом**

- Перспектива разработчика: разработчики могут извлечь выгоду из большого пула участников, которые могут помогать улучшать ПО. ПО с открытым исходным кодом также позволяет разработчикам демонстрировать свою работу

и экспертизу, и сотрудничать с другими разработчиками.

- Перспектива клиента: клиенты могут использовать и модифицировать ПО по своему усмотрению без необходимости платить лицензионные сборы. Однако клиенты могут не получать такой же уровень поддержки, как при использовании проприетарного ПО.

## **Плюсы и минусы бесплатного ПО**

- Перспектива разработчика: разработчики могут привлечь большую базу пользователей, предлагая бесплатную версию своего ПО. При этом они могут получать доход за счет рекламы, покупок в приложении или подписки на премиум-функции.

- Перспектива клиента: клиенты могут бесплатно опробовать базовую версию ПО, что может быть выгодно для тех, у кого ограничен бюджет. Однако некоторым пользователям может показаться, что им не хватает каких-то важных функций ПО и возможностей, доступных только в премиум-версии.

## **Более подробное описание каждой модели ПО**

Для читателей, у которых больше свободного времени, мы даем более обстоятельный обзор каждой модели лицензирования ПО, включая описание того, как они работают, их основные механизмы и основные положения лицензионного соглашения в каждом конкретном случае.

## **Традиционная модель лицензирования ПО**

Традиционная модель лицензирования ПО – это механизм лицензирования и получения прав на ПО, при котором ПО продается или передается клиенту на основе лицензии для установки в его собственной компьютерной системе. После заключения лицензионного соглашения и установки ПО клиент получает полный контроль над ним и несет ответственность за его обслуживание.

В этой модели лицензирования разработчики продают лицензии на программное обеспечение пользователям, как правило, в соответствии с лицензионным соглашением с конечным пользователем (End-User License Agreement, или EULA) или лицензионным соглашением на ПО. Эти соглашения определяют права и обязательства как разработчика, так и пользователя в отношении использования, распространения и модификации ПО.

Ключевые положения типичного лицензионного соглашения на программное обеспечение могут включать:

- Предоставление лицензии: в лицензионном соглашении обычно указывается объем лицензируемых прав на ПО, которое пользователь получает для его использования.
- Права интеллектуальной собственности: как правило, разработчик программного обеспечения сохраняет за собой право собственности на все права интеллектуальной собственности на ПО, включая авторские права, патенты, товарные знаки и коммерческую тайну.
- Ограничения на использование: в лицензионном соглашении могут быть указаны определенные ограничения на использование ПО, например, ограничение количества устройств, на которые может быть установлено ПО.
- Гарантия и отказ от ответственности: разработчик может отказаться от любых гарантий «товарной пригодности» ПО и дать гарантии пригодности ПО только для определенной цели, ограничив про этом свою ответственность за ущерб, возникающий в результате использования ПО. (Как правило, в лицензируемом по договору ПО не должно быть существенных ошибок, которые могут препятствовать его правильной работе, оно должно соответствовать своему назначению и выполнять базовые функции, которые пользователь может ожидать от такого типа ПО. Однако, поскольку на работу ПО может влиять ряд факторов, например, используемое оборудование, на котором оно запускается, ввод пользователей и внешние события, разработчик может быть не готовым давать гарантию на пригодность ПО для использования без каких-либо ограничений.)
- Прекращение действия: в лицензионном соглашении указывается, когда и при каких условиях действие лицензии может быть прекращено, например, в случае нарушения условий соглашения или неуплаты лицензионных сборов.

Примеры традиционных моделей лицензирования включают Microsoft Office или Adobe Creative Suite. В этой модели клиенты приобретают лицензию на определенную версию ПО и самостоятельно несут ответственность за ее установку и обслуживание.

Одним из преимуществ традиционной модели лицензирования ПО с точки зрения разработчика является бóльший контроль над ПО и потенциально более высокая прибыль по сравнению с другими моделями лицензирования. С точки зрения клиента, традиционная модель лицензирования ПО обеспечивает

большой контроль над своими данными и возможность использовать ПО даже без подключения к сети «Интернет». Тем не менее, клиенты также несут бремя поддержания ПО и обеспечения его правильной работы.

Обновления и поддержка ПО в данной модели лицензирования обычно осуществляются отдельно от первоначальной покупки лицензии на ПО. Клиентам может потребоваться приобрести дополнительные лицензии и платить за обновления, чтобы получить доступ к последней версии ПО, а поддержка может предлагаться на платной основе.

В целом, традиционная модель лицензирования может обеспечить надежный и понятный вариант как для разработчиков, так и для клиентов, но важно внимательно изучить и понять условия любых лицензионных соглашений перед заключением договора.

## **ПО как услуга (SaaS)**

Модель SaaS, или программное обеспечение как услуга, — это тип дистрибуции ПО, при котором оно размещается на сервере «в облаке» у поставщика и доступно клиенту через сеть «Интернет» на основе подписки. Клиент не владеет базовой инфраструктурой и не управляет ею, а вместо этого регулярно платит за доступ к ПО.

В модели SaaS клиенты обычно заключают с поставщиком соглашение о программном обеспечении как услуге (соглашение SaaS). Ключевые положения типичного соглашения SaaS могут включать:

- Предоставление лицензии: Соглашение SaaS обычно определяет объем и ограничения лицензии, которая предоставляется пользователю для использования ПО.
- Соглашение о минимальном уровне обслуживания (Service Level Agreement, или SLA): в соглашениях SaaS поставщики часто предлагают условия SLA, гарантирующее бесперебойную работу, производительность и доступность программного обеспечения.
- Права на интеллектуальную собственность: поставщик SaaS сохраняет за собой все права на интеллектуальную собственность в отношении ПО, включая авторские права, патенты, товарные знаки и коммерческую тайну.
- Владение данными и конфиденциальность: в соглашении SaaS может быть указано, кому принадлежат данные, которые вводятся в ПО, а также кто имеет

доступ к этим данным и как они будут использоваться. У клиентов также может быть возможность экспорта своих данных, если они решат прекратить подписку на ПО.

- Плата за подписку и условия оплаты: поставщик SaaS обычно взимает с клиентов плату на регулярной основе, ежемесячно или ежегодно. Условия оплаты, в том числе штрафы за просрочку платежа и расторжение за неуплату, также могут быть включены в соглашение.

- Прекращение: в Соглашении SaaS указывается, когда и при каких условиях подписка может быть прекращена, например, в случае нарушения условий соглашения или неуплаты абонентской платы.

Одним из преимуществ модели SaaS с точки зрения разработчика является то, что она обеспечивает стабильный поток доходов и упрощает обновление и обслуживание приложения, работающего на его собственных серверах. Эта модель также может обеспечить большую гибкость с точки зрения масштабирования и работы с большим количеством клиентов. С точки зрения клиента, нет необходимости беспокоиться об обслуживании или обновлении ПО, и клиент может получить доступ к ПО из любого места, где есть подключение к сети «Интернет». Однако у некоторых клиентов могут возникнуть опасения по поводу безопасности и конфиденциальности своих данных.

Примеры моделей SaaS включают, среди прочих, Salesforce, Slack, Dropbox и Пачку. В этой модели клиенты подписываются на услугу и получают к ней доступ через веб-браузер или специальное приложение. Поставщик несет ответственность за обновление ПО и поддержание необходимой инфраструктуры для поддержки услуги.

В целом, модель SaaS представляет собой удобный и гибкий вариант как для разработчиков, так и для клиентов, но важно внимательно изучить и понять условия любого соглашения SaaS перед подписанием.

## **Платформа как услуга (PaaS)**

Модель PaaS или платформа как услуга — это тип дистрибуции ПО, при котором поставщик предлагает платформу, позволяющую клиентам разрабатывать, запускать и управлять своими собственными приложениями. Поставщик управляет и поддерживает базовую инфраструктуру, такую как серверы, базы данных и операционные системы.



В модели PaaS клиенты обычно заключают с поставщиком соглашение об использовании платформы как услуги (соглашение PaaS). Ключевые положения типичного соглашения PaaS могут включать:

- Предоставление лицензии: соглашение PaaS обычно определяет объем и ограничения лицензии, которую пользователь получает для использования платформы и разработки собственных приложений.
- Соглашение о минимальном уровне обслуживания (Software Level Agreement, или SLA): подобно соглашениям SaaS, соглашения PaaS часто включают условия SLA, предоставляющее гарантии безотказной работы, производительности и доступности платформы.
- Права на интеллектуальную собственность: как правило, поставщик PaaS сохраняет за собой все права на интеллектуальную собственность, связанные с платформой, но может разрешить клиентам сохранять право собственности на свои собственные приложения, разрабатываемые на платформе.
- Владение данными и конфиденциальность: в соглашении PaaS может быть указано, кому принадлежат данные, введенные в приложение, и как они могут использоваться поставщиком.
- Плата за подписку и условия оплаты: как и в случае с SaaS, провайдеры обычно взимают плату с клиентов на регулярной основе, часто в зависимости от использования или выбранного плана оплаты. Условия оплаты могут варьироваться в зависимости от поставщика и потребностей клиента.
- Прекращение действия: соглашение PaaS содержит положения о том, когда и при каких обстоятельствах подписка может быть прекращена, например, в случае неуплаты, нарушения соглашения или неправильного использования платформы.

Примеры моделей PaaS включают, среди прочих, Google App Engine, Microsoft Azure и Heroku. В этой модели клиенты могут использовать инструменты, предоставляемые платформой, для создания и развертывания собственных приложений, не беспокоясь о необходимости поддержания базовой инфраструктуры.

С точки зрения клиентов-разработчиков модель PaaS позволяет снижать затраты на инфраструктуру и сосредотачиваться на разработке и развертывании своих приложений. Поставщик платформы берет на себя обязательство по обслуживанию, масштабируемости и безопасности сервера. С точки зрения

клиентов, они выигрывают от снижения затрат на IT-инфраструктуру и могут сосредоточиться на разработке собственных приложений. Однако они могут быть ограничены функциями и услугами, предоставляемыми платформой.

В целом, модель PaaS представляет собой удобный и экономичный вариант как для разработчиков, так и для клиентов, но важно внимательно изучить и понять условия любого соглашения PaaS перед подписанием.

## **Инфраструктура как услуга (IaaS)**

Модель IaaS, или инфраструктура как услуга, — это тип дистрибуции ПО, при котором поставщик предлагает вычислительную инфраструктуру, такую как виртуальные машины, хранилище и сеть, с оплатой по факту использования. Клиент контролирует операционную систему, приложения и развернутый контент, но несет ответственность за управление собственным программным обеспечением и данными.

В модели IaaS клиенты обычно заключают с поставщиком соглашение об использовании инфраструктуры как услуги (соглашение IaaS). Ключевые положения типичного соглашения IaaS могут включать:

- Предоставление лицензии: в соглашении IaaS обычно указывается объем и ограничения лицензии, которая предоставляется пользователю для использования базовой инфраструктуры.
- Соглашение о минимальном уровне обслуживания (SLA): подобно моделям SaaS и PaaS, соглашения IaaS часто включают гарантии SLA, гарантирующие бесперебойную работу, производительность и доступность инфраструктуры.
- Права на интеллектуальную собственность: как правило, поставщик инфраструктуры как услуги сохраняет право собственности на интеллектуальную собственность, связанную с базовой инфраструктурой, но заказчик сохраняет право собственности на свои собственные приложения и данные.
- Владение данными и конфиденциальность: в соглашении IaaS может быть указано, кому принадлежат данные, введенные в инфраструктуру, и как они могут использоваться поставщиком.
- Плата за подписку и условия оплаты: провайдеры обычно взимают с клиентов плату на регулярной основе в зависимости от использования, например, за время вычислений, объем хранилища и пропускную способность. Условия оплаты

могут варьироваться в зависимости от поставщика и потребностей клиента.

- Прекращение договора: в соглашении IaaS может быть указано, когда и при каких обстоятельствах подписка может быть прекращена, например, в случае неуплаты, нарушения соглашения или неправильного использования инфраструктуры.

Примеры моделей IaaS включают, среди прочего, Amazon Web Services, Microsoft Azure и Google Cloud Platform. В этой модели клиенты могут выбирать конкретные ресурсы инфраструктуры, которые им нужны, и использовать их по мере необходимости.

С точки зрения разработчика модель IaaS обеспечивает полный контроль над инфраструктурой, включая возможность настраивать и оптимизировать операционную систему и другие компоненты инфраструктуры. Эта модель также обеспечивает гибкость с точки зрения масштабирования, поскольку клиенты могут настраивать ресурсы инфраструктуры по мере необходимости. С точки зрения клиентов, они выигрывают от снижения затрат на оборудование и могут сосредоточиться на управлении собственным программным обеспечением и данными. Однако эта модель требует технических знаний и может быть дороже по сравнению с другими моделями, такими как SaaS или PaaS.

В целом, модель IaaS предоставляет широкие возможности настройки и масштабирования как для разработчиков, так и для клиентов, но важно внимательно изучить и понять условия любого соглашения IaaS перед подписанием.

## **ПО с открытым исходным кодом**

ПО с открытым исходным кодом предоставляется по лицензии с открытым исходным кодом, которая позволяет любому лицу использовать, модифицировать и заниматься дистрибуцией кода. Сообщество разработчиков работает с ПО, чтобы улучшить его, что постепенно приводит к более высокому качеству такого ПО.

В этой модели обычно нет официального договора между разработчиками и пользователями, поскольку ПО предлагается бесплатно и открыто. Вместо этого проекты с открытым исходным кодом обычно используют стандартные лицензии с открытым исходным кодом, такие как Стандартная общественная лицензия GNU (GNU GPL) или лицензия Apache (см. [нашу](#) отдельную статью по теме лицензий с открытым кодом).

Ключевые положения лицензии GNU GPL, которая является одной из наиболее широко используемых лицензий с открытым исходным кодом, включают:

- Предоставление лицензии: лицензия дает разрешение на использование, изменение и распространение ПО.
- Copyleft: это положение требует, чтобы любые изменения в ПО также были доступны всем другим лицам по той же лицензии с открытым исходным кодом.
- Права на интеллектуальную собственность: лицензия ограничивает использование проприетарных компонентов в программном обеспечении.
- Гарантии и отказ от ответственности: лицензиар отказывается от предоставления каких-либо гарантий товарной пригодности (см. выше) или пригодности для определенной цели и ограничивает свою ответственность за ущерб, возникший у лицензиата в результате использования ПО.

С точки зрения разработчиков модель лицензирования с открытым исходным кодом дает возможность продемонстрировать свою работу, сотрудничать с другими разработчиками и получать выгоду от участия большего числа разработчиков. Эта модель также может способствовать распространению ПО. С точки зрения клиента, ПО с открытым исходным кодом, как правило, бесплатное и обеспечивает большую гибкость с точки зрения изменения и настройки программного обеспечения в соответствии с потребностями пользователя. Однако пользователи могут не получать такой же уровень поддержки, как при использовании проприетарного программного обеспечения, и в таком ПО может быть не так много доступных функций по сравнению с коммерческим вариантом ПО.

Примеры ПО с открытым исходным кодом включают, среди прочих, Linux, веб-сервер Apache и базу данных MySQL. В этой модели каждый может внести свой вклад в разработку ПО и использовать его для своих целей.

В целом, модель с открытым исходным кодом обеспечивает возможность использования ПО и совместной работы над ним как для разработчиков, так и для пользователей, но важно внимательно изучить и понять условия любой лицензии с открытым исходным кодом, прежде чем использовать или вносить свой вклад в такое ПО.

## **Бесплатное программное обеспечение**

Модель программного обеспечения Freemium — это тип дистрибуции ПО, при котором базовая версия ПО предлагается бесплатно, а дополнительные функции предлагаются за определенную плату. Клиенты могут перейти на премиум-версию, заплатив единовременную плату или перейдя на подписку.

В этой модели клиенты обычно заключают соглашение о бесплатном ПО с поставщиком. Некоторые ключевые положения типичного соглашения Freemium могут включать:

- Предоставление лицензии: в соглашении обычно указывается объем и ограничения лицензии, которую пользователь получает для использования ПО.
- Права на интеллектуальную собственность: поставщик сохраняет за собой все права на интеллектуальную собственность в отношении ПО, включая авторские права, патенты, товарные знаки и коммерческую тайну.
- Сборы и условия оплаты: поставщики могут взимать плату за премиум-функции или предлагать подписку в зависимости от использования.
- Владение данными и конфиденциальность: в соглашении может быть указано, кому принадлежат данные, введенные в ПО, и как они могут использоваться поставщиком.
- Прекращение договора: в соглашении может быть указано, когда и при каких обстоятельствах подписка может быть прекращена, например, в случае неуплаты или неправильного использования ПО.

С точки зрения разработчика, модель Freemium позволяет предлагать бесплатную версию своего ПО большому количеству пользователей. Это, в свою очередь, может привести к более широкому внедрению, повышению узнаваемости бренда и возможности продавать дополнительные функции или услуги премиального уровня. При этом разработчики могут получать доход за счет рекламы, покупок в приложении или подписки на премиум-функции.

С точки зрения клиентов, они могут бесплатно опробовать базовую версию ПО, что может быть выгодно тем, у кого ограничен бюджет. Однако некоторым пользователям может показаться, что им не хватает важных функций ПО, которые доступны только в премиум-версии, и также могут возникнуть проблемы с безопасностью и конфиденциальностью данных.

Примеры бесплатных программных моделей лицензирования включают, помимо прочих, Dropbox, LinkedIn и Spotify.

В целом модель Freemium предоставляет гибкий и масштабируемый вариант как для разработчиков, так и для пользователей, но важно внимательно изучить и понять условия соглашения Freemium перед использованием ПО. Кроме того, клиенты должны знать о потенциальных ограничениях, связанных с использованием бесплатной версии ПО, таких как ограниченное хранилище, реклама или ограниченные функции.

## Заключение

Как указано выше, разработчики и заказчики (пользователи) могут выбирать различные модели лицензирования и использования программного обеспечения. Модель SaaS предлагает удобство и доступность, PaaS предоставляет платформу для разработки и развертывания ПО, IaaS обеспечивает разработчикам полный контроль над инфраструктурой, ПО с открытым исходным кодом обеспечивает совместную работу и гибкость, а модель Freemium предоставляет гибкий и масштабируемый вариант как для разработчиков, так и для пользователей-клиентов. Каждая модель имеет свои плюсы и минусы. И разработчикам, и клиентам важно внимательно изучить и понять условия соглашения о лицензировании, прежде чем останавливать свой выбор на конкретной модели лицензирования или использования ПО. В конечном итоге, решение должно основываться на индивидуальных потребностях, бюджете и наличии должного уровня технической экспертизы, а также на конкретных целях и требованиях к распространяемому ПО.

\* \* \*

Этот материал предназначен для общей информации и не является юридической консультацией. Если у Вас есть какие-либо вопросы или Вы хотите узнать больше о моделях лицензирования ПО, направляйте [нам](#) свои запросы по адресу электронной почты [info@danilovpartners.ru](mailto:info@danilovpartners.ru).