



ОТКРЫТОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

Сборник мануалов

Оригинал статей: opensource.guide





Открытое программное обеспечение создано такими же людьми, как и вы. С помощью этих статей вы научитесь создавать и продвигать свои собственные проекты.

[Lemma Works](#)

[Ветка на Exploit](#)

[Ветка на OldDot](#)

СОДЕРЖАНИЕ:

КРАТКИЙ ГИД ПО РАЗДЕЛАМ.....	4
1. КАК ВНЕСТИ СВОЙ ВКЛАД	5
ЗАЧЕМ ЭТО НУЖНО?	5
<i>Развитие существующих навыков</i>	<i>6</i>
<i>Знакомство с людьми со схожими интересами</i>	<i>6</i>
<i>Поиск наставников и учеников</i>	<i>6</i>
<i>Способ приобрести репутацию и, возможно, карьеру.....</i>	<i>6</i>
<i>Обретение социальных навыков</i>	<i>6</i>
<i>Возможность вносить любые, даже маленькие поправки.....</i>	<i>6</i>
ЧТО ИМЕЕТСЯ В ВИДУ ПОД СЛОВАМИ «ВНЕСТИ СВОЙ ВКЛАД»?	6
<i>Не обязательно уметь писать код</i>	<i>7</i>
<i>Вам нравится планировать и организовывать какие-либо события?</i>	<i>7</i>
<i>Предпочитаете работать с дизайном?</i>	<i>7</i>
<i>Работаете с документацией и текстами?.....</i>	<i>8</i>
<i>Нравится решать организационные вопросы?.....</i>	<i>8</i>
<i>Любите работать с кодом?.....</i>	<i>8</i>
<i>Хотите просто помогать людям?</i>	<i>8</i>
<i>Нравится помогать другим участникам с кодом?.....</i>	<i>8</i>
<i>Не нужно упираться в проекты программного обеспечения!.....</i>	<i>9</i>
ОРИЕНТИРУЯСЬ НА НОВЫЙ ПРОЕКТ	9
<i>Анатомия проекта с открытым исходным кодом</i>	<i>9</i>
ВЫБОР ПРОЕКТА ДЛЯ УЧАСТИЯ	11
<i>На что стоит обратить внимание?</i>	<i>11</i>
ПРАВИЛЬНОЕ ВНЕСЕНИЕ ВКЛАДА В ПРОЕКТ.....	13
<i>Выстраивать продуктивное общение.....</i>	<i>13</i>
<i>Сбор контекста</i>	<i>14</i>
<i>Открытие вопроса (проблемы)</i>	<i>15</i>
<i>Открытие запроса на принятие изменений</i>	<i>16</i>
ЧТО ДАЛЬШЕ?	16
 <i>Никакой реакции.....</i>	<i>17</i>
 <i>Кто-то попросит внести изменения</i>	<i>17</i>
 <i>Вам откажут в принятии вашей работы.....</i>	<i>17</i>
 <i>Ваш вклад будет принят.....</i>	<i>17</i>
Вы СДЕЛАЛИ ЭТО!	18
2. НАЧАЛО РАБОТЫ С ПРОЕКТОМ	19
«ЧТО» И «ПОЧЕМУ» OPENSOURCE-ПРОЕКТОВ.....	19
<i>Что означает «открытый исходный код»?.....</i>	<i>19</i>
<i>Почему люди открывают исходный код своей работы?</i>	<i>20</i>
<i>Означает ли «opensource» «бесплатно»?.....</i>	<i>20</i>
НУЖНО ЛИ МНЕ ЗАПУСКАТЬ СОБСТВЕННЫЙ OPENSOURCE-ПРОЕКТ?	21
<i>Постановка целей</i>	<i>21</i>
<i>Участие в других проектах.....</i>	<i>22</i>
ЗАПУСК СОБСТВЕННОГО ПРОЕКТА С ОТКРЫТЫМ ИСХОДНЫМ КОДОМ	22
<i>Выбор лицензии</i>	<i>23</i>
<i>Написание README.....</i>	<i>23</i>
<i>Написание руководств по вкладам</i>	<i>24</i>
<i>Разработка правил поведения</i>	<i>25</i>
НАЗВАНИЕ И БРЕНДИНГ ВАШЕГО ПРОЕКТА	26
<i>Выбор правильного имени</i>	<i>26</i>
<i>Как избежать конфликтов имён</i>	<i>26</i>
<i>То, как вы подаёте информацию (даже код), влияет на бренд!</i>	<i>26</i>

КОНТРОЛЬНЫЙ СПИСОК ПЕРЕД ЗАПУСКОМ	27
Ты СДЕЛАЛ ЭТО!.....	28
3. ПОИСК ПОЛЬЗОВАТЕЛЕЙ ВАШЕГО ПРОЕКТА	29
РАСПРОСТРАНЯЙТЕ ИНФОРМАЦИЮ	29
СФОРМУЛИРУЙТЕ СВОЁ ПОСЛАНИЕ.....	29
Люди должны СУМЕТЬ НАЙТИ И СЛЕДОВАТЬ ВАШЕМУ ПРОЕКТУ.....	30
ОНЛАЙН-ОБЩЕНИЕ.....	31
ОФФЛАЙН-ОБЩЕНИЕ	32
СОЗДАВАЙТЕ РЕПУТАЦИЮ	33
ДЕРЖИТЕСЬ! НЕ СДАВАЙТЕСЬ!.....	34
4. ОРГАНИЗАЦИЯ ДРУЖЕЛЮБНОГО СООБЩЕСТВА.....	35
СОЗДАНИЕ УСЛОВИЙ ДЛЯ УСПЕШНОГО ПРОЕКТА	35
<i>Заставьте людей чувствовать себя желанными</i>	<i>35</i>
<i>Документируйте всё!.....</i>	<i>37</i>
<i>Будьте отзывчивы.....</i>	<i>38</i>
<i>Предоставьте сообществу возможность общаться.....</i>	<i>38</i>
РАЗВИТИЕ СООБЩЕСТВА	39
<i>Не проявляйте излишней толерантности</i>	<i>39</i>
<i>Приветствуйте новых участников.....</i>	<i>40</i>
<i>Разделите управление проектом</i>	<i>41</i>
РАЗРЕШЕНИЕ КОНФЛИКТОВ.....	42
<i>Установите «планку» для доброты.....</i>	<i>42</i>
<i>Относитесь к README как к конституции</i>	<i>43</i>
<i>Основной фокус – на путешествии, а не на конечном пункте</i>	<i>43</i>
<i>Не позволяйте разговорам уходить в сторону</i>	<i>44</i>
<i>Готовьтесь с «сражениям» с умом.....</i>	<i>44</i>
<i>Определить «арбитра» сообщества.....</i>	<i>44</i>
СООБЩЕСТВО – ЭТО СЕРДЦЕ ОТКРЫТЫХ ПРОЕКТОВ.....	45
5. СОВЕТЫ ПО СОПРОВОЖДЕНИЮ ПРОЕКТА.....	46
ЧТО ЗНАЧИТ БЫТЬ СОПРОВОДИТЕЛЕМ ПРОЕКТА?	46
ДОКУМЕНТИРОВАНИЕ ПРОИСХОДЯЩЕГО.....	46
<i>Запишите ваше видение проекта</i>	<i>47</i>
<i>Сообщите людям свои ожидания.....</i>	<i>47</i>
<i>Не уводите общение в приват</i>	<i>48</i>
УЧИТЕСЬ ГОВОРИТЬ «НЕТ»	48
<i>Оставайтесь доброжелательны</i>	<i>48</i>
<i>Будьте инициативны</i>	<i>50</i>
<i>Практикуйте наставничество.....</i>	<i>50</i>
ИСПОЛЬЗУЙТЕ ВАШЕ СООБЩЕСТВО	51
<i>Поделитесь работой.....</i>	<i>51</i>
<i>Позвольте людям создавать решения, которые им нужны</i>	<i>51</i>
ИСПОЛЬЗУЙТЕ РОБОТОВ	52
<i>Проводите тесты и прочие проверки – улучшайте качество кода</i>	<i>52</i>
<i>Автоматизируйте основные задачи обслуживания</i>	<i>52</i>
ДЕЛАТЬ ПАУЗУ – ЭТО НОРМАЛЬНО!	53
ГЛАВНОЕ – ПОЗАБОТЬТЕСЬ О СЕБЕ!.....	54
6. ЛИДЕРСТВО И УПРАВЛЕНИЕ	55
РАЗБИРАЕМСЯ С УПРАВЛЕНИЕМ ВАШЕГО РАСТУЩЕГО ПРОЕКТА.....	55
КАКИЕ РОЛИ ПРИНИМАЮТ ЛЮДИ ОТКРЫТЫХ ПРОЕКТАХ?	55
КАК ОФИЦИАЛЬНО ОФОРМИТЬ ЭТИ РОЛИ?.....	56
КОГДА СТОИТ ПРЕДОСТАВИТЬ КОМУ-НИБУДЬ ДОСТУП К КОММИТАМ?	57

ТИПОВЫЕ СТРУКТУРЫ УПРАВЛЕНИЯ ДЛЯ OPENSOURCE-ПРОЕКТОВ.....	58
Нужна ли документация по управлению при запуске проекта?	58
Что если сотрудники компании начнут вносить вклады?	59
Нужно ли юридическое лицо для поддержки моего проекта?	59
7. ПОЛУЧЕНИЕ ПРИБЫЛИ ОТ ПРОЕКТА.....	61
Почему некоторые люди ищут финансовую поддержку	61
Получение оплаты за своё время	63
Поиск финансирования для вашего проекта.....	64
<i>Краудфандинговые кампании или спонсорство</i>	64
<i>Поток дохода</i>	64
<i>Грантовое финансирование</i>	65
Аргументация для получения финансовой поддержки.....	65
<i>Влияние</i>	65
<i>История</i>	65
<i>Выгода спонсора</i>	65
<i>Использование спонсорских средств</i>	66
<i>Получение средств</i>	66
Экспериментируйте и не сдавайтесь.....	66
8. КОДЕКС СООБЩЕСТВА	67
Зачем вообще нужны правила поведения?	67
Разработка кодекса поведения	67
Решите, как будет применяться кодекс поведения	68
Обеспечение соблюдения вашего кодекса поведения.....	68
<i>Соберите полную информацию о ситуации</i>	69
<i>Примите соответствующие меры</i>	69
Ваши обязанности в качестве сопровождающего проекта.....	70
Поощряйте поведение, которое вы хотите видеть вокруг себя.....	70
9. МЕТРИКИ ПРОЕКТОВ	71
Зачем вообще что-то измерять?.....	71
Как люди находят ваш проект?	72
Использование проекта	72
Сохранение пользователей	73
Деятельность сопровождающих.....	74
Используйте метрики, чтобы узнать больше о людях	75
10. ЮРИДИЧЕСКАЯ СТОРОНА ОТКРЫТОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ.....	76
Разбираемся в правовых моментах.....	76
Почему людей так волнует юридическая сторона проектов?	76
Являются ли публичные проекты GitHub OPENSOURCE-ПРОЕКТАМИ?	77
Просто дайте мне всё для защиты проекта без TL;DR	78
Какая лицензия подходит для моего проекта?	78
Что если я хочу изменить лицензию на мой проект?	79
Нужны ли моему проекту дополнительные соглашения?	80
Что должен знать юридический отдел моей компании?	81

Краткий гид по разделам

- Как внести свой вклад

Хотите внести свой вклад в разработку открытого программного обеспечения? Этот раздел – руководство, которое поможет и новичку, и опытному разработчику.

- Начало работы с проектом

Основная информация об открытом программном обеспечении и базовая подготовка к запуску своего проекта.

- Поиск пользователей

В данном разделе мы разберёмся, как «взрастить» свой проект, привлекая к нему пользователей.

- Организация сообщества

Необходимые знания для создания сообщества вокруг вашего проекта, или как мотивировать людей использовать вашу разработку, участвовать в развитии и распространении проекта.

- Советы по сопровождению проекта

В этом разделе мы разберёмся, как максимально упростить процесс сопровождения программного обеспечения с открытым исходным кодом – начиная от создания документации к процессам и работы с сообществом.

- Лидерство и управление

Основные правила принятия решений, которые помогут вам активно развивать свой проект.

- Получение прибыли от проекта

Получение финансовой поддержки для дальнейшего развития вашего проекта.

- Кодекс сообщества

Этот раздел расскажет о том, как добиться конструктивного и адекватного поведения членов сообщества с помощью кодекса поведения сообщества.

- Метрики проектов

Руководство по грамотному отслеживанию метрик проекта – это поможет вам принять верные и обоснованные решения по развитию вашего проекта.

- Юридическая сторона открытого программного обеспечения

В этом разделе вы найдёте всю важную информацию о юридическом подтексте проектов с открытым исходным кодом – всё что вы хотели узнать и ещё кое-что 😊

1. Как внести свой вклад

Хотите внести свой вклад в разработку открытого программного обеспечения? Этот раздел – руководство, которое поможет и новичку, и опытному разработчику.



Зачем это нужно?



Работа с freenode помогла мне получить многие навыки, которые понадобились мне как для учёбы в университете, так и для дальнейшей работы. Я уверен, что моё участие в разработке проектов помогает и мне, и непосредственно сообществу!

@errietta

Внесение своего вклада в проекты с открытым исходным кодом – довольно полезный способ обучения и накопления опыта в практически любых навыках, которые только можно представить. И всё же, зачем именно люди занимаются этим? Давайте рассмотрим несколько причин!

Развитие существующих навыков

Неважно, чем вы занимаетесь: написание кода, UI-дизайн, графический дизайн, организация кампаний – если вам нужна дополнительная практика, тогда открытие своего проекта станет для вас полезнейшим опытом.

Знакомство с людьми со схожими интересами

Проекты с открытым исходным кодом очень часто окружены тёплыми, гостеприимными сообществами, в которых люди знакомы на протяжении долгих лет. В таких сообществах рождается множество крепких дружеских отношений, причём неважно как эти люди знакомятся – в процессе совместной работы, на конференциях или просто в ночных чатах о буррито 😊

Поиск наставников и учеников

Работая с людьми, вам зачастую придётся и объяснять, что и как вы делаете, и спрашивать совета или помощи. Такое общение безусловно полезно для всех участников проектов.

Способ приобрести репутацию и, возможно, карьеру

Фактически, все проекты с открытым исходным кодом являются общедоступными, так что у вас всегда будет под рукой неплохое портфолио из собственных работ.

Обретение социальных навыков

Работа над подобными проектами часто требует лидерских и управленческих навыков: организация команды, разрешение конфликтов и расстановки правильных приоритетов; нет сомнений, что всё это пригодится вам позже.

Возможность вносить любые, даже маленькие поправки

Для того, чтобы получать удовольствие от работы с проектами с открытым исходным кодом, вам не нужно становиться пожизненным официальным участником. Вы когда-нибудь встречали на сайте опечатки? Возникало желание исправить это? Так вот – в открытых проектах каждый может вносить такие поправки – люди могут самолично влиять на окружающий их мир, и это само по себе приятно.

Что имеется в виду под словами «внести свой вклад»?

Если вы новичок в этой сфере, то это может выглядеть пугающе. Как найти подходящий проект? Что делать, если вы не знаете, как писать код? Вдруг что-то пойдёт не так?

Не нужно беспокоиться! Существует множество способов поучаствовать в проекте с открытым исходным кодом; вот несколько советов, которые помогут вам получить максимальный результат от этого опыта.

Не обязательно уметь писать код

Это довольно распространённое заблуждение. На деле же, есть куча других сфер деятельности, которые часто [игнорируются многими участниками](#). Таким образом, работая в этих сферах, вы можете оказать проекту неоценимую помощь!



Меня многие знают по моей работе над CocoaPods, но большинство из них даже не подозревает, что фактически я не работаю непосредственно над инструментом CocoaPods. В основном я трачу время на подготовку документации и брендинг.

@orta

Даже в том случае если вы предпочитаете работу с кодом, работа над другими сферами проекта – отличный способ помочь развитию этого проекта и начать общаться с другими членами сообщества.



Первое моё общение с командой разработчиков Python (python-dev) состоялось после того, как в июне 2002 года я сделал рассылку со своим патчем к проекту. Я нашёл в коде ошибку и отправил им поправку; они быстро отреагировали и запросили дополнительных пояснений – это много для меня значило. Но гораздо более важный для меня фактор – это то, что я лично смог что-то исправить в готовом проекте.

@brettcannon

Вам нравится планировать и организовывать какие-либо события?

- Организуйте в рамках проекта семинары или встречи (к примеру, как делал [@fzamperin для NodeSchool](#));
- Управляйте проектной конференцией (если она есть);
- Помогайте членам сообщества: подбирайте для них нужные конференции и предлагайте темы для выступления.

Предпочитаете работать с дизайном?

- Работайте над макетами и интерфейсом – сделайте проект комфортным в использовании;
- Проводите опросы пользователей – узнавайте, как именно стоит реорганизовать или изменить навигацию или меню проекта (здесь как пример можно привести работу , как предлагает [Drupal](#));
- Составьте руководство по стилям – проект должен иметь последовательный визуальный дизайн;
- Создавайте рисунки для футболок или логотипы (пример – [участники hapi.js](#)).

Работаете с документацией и текстами?

- Создавайте и правьте документацию по проекту;
- Создайте портфолио с примерами использования проекта;
- Занимайтесь рассылкой новостей проекта или курируйте тематики для рассылки;
- Напишите мануалы по проекту (обратите внимание на работу участников [PyPA](#));
- Займитесь переводом документации проекта.



На самом деле, документация - крайне важна. Если задуматься, то именно она и стала великолепной отличительной чертой проекта [Babel](#). Есть разделы, которые можно наполнять примерами своих работ, и даже банальное добавление параграфа чрезвычайно ценится.

@kittens

Нравится решать организационные вопросы?

- Создавайте ссылки на решение проблем, уже встречавшихся ранее и предлагайте систему упорядочивания запросов;
- Просматривайте открытые запросы и закрывайте решённые (как [@nzakas для ESLint](#));
- Задавайте уточняющие вопросы по нерешённым запросам, помогайте развиваться дискуссии).

Любите работать с кодом?

- Ищите проблемы, которые вы можете решить (как [@dianjin для Leaflet](#));
- Предложите помощь в написании новой функции;
- Автоматизируйте настройку проекта;
- Работайте над улучшением инструментов и тестированием.

Хотите просто помогать людям?

- Отвечайте на вопросы о проекте, (например, как в [этом примере Postgres](#)) или на Reddit;
- Отвечайте на вопросы людей по нерешённым вопросам;
- Помогайте с модерацией форумов или чатов.

Нравится помогать другим участникам с кодом?

- Изучайте код других участников;
- Создавайте мануалы по использованию проекта;
- Предлагайте наставникам участников (как [@ereichert для @bronzdoc](#)).

Не нужно упираться в проекты программного обеспечения!

Хотя «открытый исходный код» чаще всего относится к программному обеспечению, работать можно над чем угодно. Есть открытые проекты книг, сборников рецептов, списков, классов и много-много другого.

Например:

- **@sindresorhus** курирует [список «потрясающих» топикиов](#);
- **@h5bp** работает над составлением [списком вопросов для собеседований](#) кандидатов на должность front-end разработчиков;
- **@stuartlynn** и **@nicole-a-tesla** собирали [коллекцию забавных фактов о тупиках](#).

Даже если вы разработчик программного обеспечения, работа над проектной документацией может помочь вам начать работу в данной сфере. Часто бывает, что проекты, в которых не используется код, кажутся менее пугающими, а процесс работы над такими проектами и сотрудничество поможет в создании репутации и наработке опыта.

Ориентируясь на новый проект



Если вы смотрите на запросы в мониторинге проблем и всё кажется запутанным, проблема не только в вас. Работа с этими инструментами требует хорошего уровня знаний, но люди могут помочь вам разобраться – не стесняйтесь задавать им вопросы.

@shaunagm

Если участие в открытом проекте заключается в чём-то большем, чем просто исправление опечаток, то начало работы в коллективе похоже на разговор с группой незнакомцев на вечеринке. Если вы начнёте говорить о ламах, а они вели пространную беседу о золотых рыбках, они, вероятно, будут смотреть на вас немного странно.

Прежде чем врываться в обсуждение с ворохом собственных предложений, стоит начать с чтения топикиов и изучения правил обсуждения. Такой подход увеличивает шансы, что ваши идеи будут замечены и услышаны.

Анатомия проекта с открытым исходным кодом

Каждое сообщество подобных проектов уникально.

Потратив годы на работу с одним проектом с открытым исходным кодом, вы освоите только этот конкретный проект. Начав работу с другим проектом, вы столкнётесь с абсолютно другой терминологией, нормами и стилем общения.

Несмотря на это, многие проекты имеют схожую структуру. Если вы в курсе основных ролей в сообществе и представляете, как будет проходить работа, то вы сможете быстро сориентироваться в любом новом проекте.

Обычно в проекте люди занимают следующие роли:

- Автор: человек (люди) или организация, которые создали проект;

- Владелец: лицо или лица, являющиеся официальными собственниками репозитория или проекта (и это не всегда автор);
- Сопроводители проекта: участники, ответственные за внешний вид и организационные аспекты проекта (они также могут быть авторами или владельцами проекта);
- Участники: каждый, кто внёс свой вклад в проект;
- Члены сообщества: люди, которые используют проект. Они участвуют в обсуждениях и могут свободно высказать своё мнение о развитии проекта.

В более крупных проектах также могут присутствовать «подгруппы» или рабочие группы, ориентированные на различные задачи (к примеру, разработка инструментов, управление системами, модерация сообщества и организация мероприятий). Обычно информация о команде проекта расположена на соответствующей странице сайта проекта.

Также у любого проекта есть документация. Обычно её можно найти в самом верху репозитория:

- **ЛИЦЕНЗИЯ (LICENSE):** по определению, у каждого проекта с открытым исходным кодом должна быть такая же лицензия. Если таковая отсутствует, то это не opensource-проект;
- **О ПРОЕКТЕ (README):** это своего рода инструкция, приветствующая новых участников сообщества и проекта. Данный документ объясняет, чем полезен проект и как начать работу;
- **УЧАСТИЕ (CONTRIBUTING):** README помогает пользователям разобраться в проекте, а этот документ разъясняет как внести в проект свой вклад (процесс участия, сферы работы и так далее). Такой документ есть не у каждого проекта, и его наличие сигнализирует о том, что участие в разработке проекта приветствуется;
- **ПРАВИЛА ПОВЕДЕНИЯ (CODE_OF_CONDUCT):** здесь можно найти основные правила поведения участников. Эти рамки способствуют формированию дружного и гостеприимного сообщества. Опять же, правила есть не у каждого проекта;
- **Прочая документация (other documentation):** какая-либо дополнительная информация, к примеру – учебные пособия, пошаговые руководства или политики управления (особенно часто встречается в крупных проектах).

Ещё один важный момент – знать, какие инструменты используются для управления обсуждениями. Изучив архивы, вы получите довольно полное представление о том, как сообщество мыслит и как оно работает.

- **Мониторинг проблем:** здесь люди обсуждают вопросы, связанные с проектом;
- **Запросы на принятие изменений:** здесь люди обсуждают и тестируют изменения в проекте;
- **Дискуссионные форумы или списки рассылки:** в некоторых проектах эти каналы используются для обсуждений вместо отчётов об ошибках или функции запросов дополнительных опций. Прочие проекты используют мониторинг проблем для всех обсуждений.
- **Чат-канал:** некоторые проекты используют чаты (к примеру, Slack или IRC) для повседневного общения, совместной работы и быстрого обмена данными.

Выбор проекта для участия

Теперь, когда мы разобрались с основами, пришло время найти проект, в который вы можете внести свой вклад!

Если вы никогда раньше не участвовали в opensource-проектах, воспользуйтесь советом Джона Кеннеди, который однажды сказал: «Не спрашивайте, что ваша страна может сделать для вас, – спросите, что вы можете сделать для своей страны».

Внести свою лепту в любой проект можно на любом уровне и этапе работы. Не стоит задумываться о том, каким именно будет первый вклад или как он будет выглядеть – вместо этого просто обратите внимание на проекты, которыми вы уже пользуетесь или хотели бы пользоваться. Как правило, самые ценные вклады делают люди, которые используют проект лично.

Если вы поймаете себя на мысли, что можно что-то улучшить или сделать иначе – доверьтесь своим инстинктам: opensource-проекты — это не какой-либо закрытый клуб для узкого круга людей; в этом проекте каждая, даже самая маленькая поправка или байт кода созданы такими же людьми, как и вы. «Проект с открытым исходным кодом» — это просто причудливый термин, который даёт понять: любая ошибка поправима.

Вы можете изучить README-файл и найти неработающую ссылку или опечатку; вы можете обратить внимание, что что-то не работает, или найти любые возможные несоответствия в документации. Вместо того, чтобы игнорировать эти моменты или искать того, кто это поправит, вы можете сами разобраться с этой проблемой, ведь это и есть суть проектов с открытым исходным кодом!

Интересный факт: 28 процентов спонтанных поправок в проектах – это работа с документацией (исправление опечаток, форматирование или добавление переводов).

Для поиска проектов можно использовать следующие ресурсы:

- [GitHub Explore](#)
- [Open Source Friday](#)
- [First Timers Only](#)
- [CodeTriage](#)
- [24 Pull Requests](#)
- [Up For Grabs](#)
- [Contributor-ninja](#)
- [First Contributions](#)
- [SourceSort](#)

На что стоит обратить внимание?

Найдя проект, в котором вы бы хотели принять участие, следует убедиться, что он подходит для этого. В противном случае, ваша работа может оказаться бесполезной.

Ниже мы привели удобный список пунктов для оценки (его можно даже распечатать ☺):

Соответствует определению проекта с открытым исходным кодом:

- ✓ Есть ли у него лицензия? Обычно она находится в корне репозитория (LICENSE).

Проект готов принимать вклады участников:

Обратите внимание на активность в основной ветке. К примеру, на GitHub эту информацию можно найти на домашней странице репозитория.

- ✓ Время последнего коммита (commit – зафиксированное изменение);
- ✓ Количество участников в проекте;
- ✓ Частота и периодичность коммитов (на GitHub это можно посмотреть во вкладке «Commits».)

Далее стоит изучить проблемные сферы проекта:

- ✓ Сколько на данный момент открытых вопросов?
- ✓ Насколько быстро реагируют сопровождающие проекта?
- ✓ Активно ли обсуждаются проблемы?
- ✓ Какого рода вопросы сейчас в стадии решения?
- ✓ Решаются ли проблемы? (На GitHub это можно узнать из вкладки «closed» страницы «Issues»).

Теперь так же проверим запросы на принятие изменений:

- ✓ Сколько на данный момент запросов?
- ✓ Быстро ли сопровождающие реагируют на запросы?
- ✓ Активно ли обсуждаются запросы?
- ✓ Есть ли запросы, размещённые недавно?
- ✓ Как давно были удовлетворены какие-либо запросы? (На GitHub эта информация находится на вкладке «closed» страницы «Pull Requests»).

Ну и наконец, проверим признаки дружелюбности проекта:

Хороший, дружелюбный проект соответствует ещё нескольким критериям. Стоит задать ещё несколько вопросов.

- ✓ Помогают ли сопровождающие людям, задающим вопросы в обсуждении проблем?
- ✓ Дружелюбны ли люди в обсуждениях проблем, на дискуссионном форуме и в чате (к примеру IRC или Slack)?
- ✓ Рассматриваются ли запросы на принятие изменений?
- ✓ Благодарят ли сопровождающие проекта участников за их вклад?



Когда вы сталкиваетесь с длинной веткой, обратите внимание на комментарии основных разработчиков. Конструктивно ли они суммируют информацию, вежливо ли подводят сообщество к решению? Если в ветке господствуют холивары, это часто является признаком того, что вся энергия уходит в споры, а не в развитие проекта.

@kfogel

Правильное внесение вклада в проект

Итак, вы нашли проект, который вам нравится, и жаждете внести в него свой вклад. Наконец-то! Как же сделать это правильно?

Выстраивать продуктивное общение

Неважно, будете ли вы принимать участие в этом проекте лишь раз или надеетесь стать частью сообщества, работа с другими людьми – это один из самых важных навыков, приобретаемый в процессе работы с проектом.



Будучи новичком, я довольно быстро поняла, что мне нужно задавать вопросы, если я хочу иметь возможность решить проблему. Сначала я пролистала основные материалы проекта, и, как только более-менее разобралась в происходящем – начала задавать вопросы. И – всё оказалось очень просто – получив необходимые пояснения я легко решила свои вопросы!

@shubheksha

Перед тем, как задать вопрос в чате, выложить запрос на принятие изменений или запросить решение проблемы, учтите несколько моментов:

Предоставьте контекст. Людям куда проще разобраться при его наличии. Если вы столкнулись с ошибкой, постарайтесь объяснить, что вы пытались сделать и как её воспроизвести. Если вы предлагаете новую идею, объясните, почему вы считаете, что она будет полезна для проекта (а не только для вас лично!)

😏 «X не происходит, когда я делаю Y»

😞 «X сломан! Пожалуйста, исправьте это».

Постарайтесь заранее сделать наработки. Не страшно, если не получится – вы хотя бы попытались. Прежде чем обращаться за помощью, обязательно ознакомьтесь с README-файлом проекта, документацией, открытыми и закрытыми проблемами, поищите решение в Интернете. Люди оценят, если увидят, что вы проявляете желание учиться.

😏 «Я не уверен, как правильно реализовать X. Я изучил всю доступную информацию и не нашёл правильного решения».

😞 «Как мне сделать X?»

Ваши запросы должны быть короткими и ясными. Так же, как с электронными письмами, каждый ваш вклад, независимо от того, насколько он прост или полезен, требует чужого внимания. Во многих проектах поступающих запросов больше, чем людей, которые готовы и могут помочь. Будьте лаконичны. Так гораздо больше шансов, что кто-то сможет вам помочь.

😏 «Я хотел бы написать руководство по API».

😬 «На днях я ехал по шоссе и у меня кончился бензин, а потом у меня появилась эта удивительная идея нашего дальнейшего развития. Но, прежде чем я вам её распишу, позвольте мне показать вам...»

Общение должно быть публичным. Хотя это и заманчиво, старайтесь не обращаться к сопровождаителям проекта в частном порядке (разумеется, кроме тех случаев, когда вам нужно поделиться конфиденциальной информацией, к примеру по вопросам безопасности или серьёзного нарушения поведения). Если вы ведёте разговор на виду у всех, многие люди смогут в будущем разобраться с подобной проблемой или вынести что-то важное из вашей беседы. Такие обсуждения – сами по себе хороший вклад в проект.

🗨️ (комментарием) «@-maintainer Привет! Что мы будем делать с моим запросом дальше?»

✉️ (по электронной почте) «Привет, извините, что беспокою, но мне интересно, была ли у вас возможность посмотреть мой запрос?»

Задавать вопросы – это нормально (однако, имейте терпение!). Каждый в сообществе когда-то был новичком в этом проекте, и даже опытным участникам нужно время, чтобы влиться в новый проект. К тому же, даже те, кто сопровождает проект уже долгое время, не всегда могут ответить на конкретно этот вопрос. Будьте терпеливы, и люди к вам потянутся 😊

😊 «Спасибо за то, что уделили время моей проблеме. Я попробовал сделать так, как вы советовали, и вот что получилось.»

😬 «Почему вы не можете решить мою проблему? Разве это не ваш проект?»

Уважайте решения сообщества. Ваши идеи могут идти вразрез с приоритетами сообщества или их видением проекта. Они могут как оставить положительный отзыв, так и решить отказаться от пути, который вы предложили. На самом деле, сопровождаителям проекта придётся работать с результатами вашего решения дольше вас, так что стоит обсуждать эти решения и искать компромиссы. Если же вы категорически не согласны с направлением развития, которое было выбрано сообществом, вы всегда можете разработать собственный форк или основать собственный проект.

😊 «Я разочарован тем, что вы не можете поддержать мою идею, но понимаю, почему она не подходит большинству пользователей. Спасибо за внимание.»

😬 «Почему вы зарубили эту идею? Это, мать его, неприемлемо!»

И прежде всего, будьте выше негатива. Над открытыми проектами работают люди со всего мира. Контекст может теряться из-за разницы языков, культур, географического расположения и часовых поясов. Также стоит учитывать, что и тон, и настроение довольно сложно передать письменной речью. Будьте настроены положительно – отклоняйте идеи вежливо, умейте вежливо задать вопрос и попросить уточнения. Просто попытайтесь сделать так, что Интернет станет чуточку лучшим местом.

Сбор контекста

Прежде чем предпринимать какие-либо действия, убедитесь, что ваша идея или вопрос не обсуждались где-то ещё. Изучите README проекта, уделите время открытым и закрытым

вопросам и запросам, загляните в [Stack Overflow](#). Не нужно тратить часы и дни, чтобы разораться абсолютно во всём, но быстрый поиск по нескольким ключевым терминам тоже даст хороший результат.

Если вы нигде не нашли подобной идеи – самое время сделать шаг. Если проект расположен на GitHub, то вы, вероятнее всего, будете общаться, открывая вопрос или размещая запросы на принятие изменений:

Открытие вопроса (проблемы) – чем-то похоже на начало разговора или обсуждения;

Запросы на принятие изменений – это начало работы над решением вопроса;

Для простого общения, к примеру, для получения какого-то уточнения или незначительного вопроса, попробуйте задать вопрос на Stack Overflow, в IRC, Slack или других каналах чата, если таковые имеются у проекта.

Перед тем, как открыть вопрос или запрос на принятие изменений, стоит уделить время тому, чтобы изучить документацию по внесению вкладов участниками (обычно такую информацию можно найти в файле CONTRIBUTING или README) – так вы узнаете, как сделать это правильно. К примеру, может быть необходимо следовать определённому шаблону или выложить результаты определённых тестов.

Если вы хотите внести существенный вклад, сначала стоит осведомиться, нужен ли он. Полезно понаблюдать за проектом некоторое время (на GitHub есть возможность настроить уведомления обо всех событиях) и узнать больше о сообществе.



Вы можете многому научиться, просто следя за интересующим вас проектом на GitHub и внимательно изучая каждый вопрос и запросы на принятие изменений.

@gaelaron

Открытие вопроса (проблемы)

Обычно это делается в следующих ситуациях:

- Чтобы сообщить об ошибке, которую вы не можете решить самостоятельно;
- Чтобы обсудить обширную тему или идею (например, касательно сообщества, видения проекта или политики проекта);
- Чтобы предложить новую функцию или другую идею для проекта.

Советы по выстраиванию общения:

- **Если вы видите открытый вопрос, который вы хотели бы решить**, оставьте комментарий, чтобы люди знали, что вы возьмётесь за него. Таким образом, люди не потратят своё и ваше время, дублируя вашу работу;
- **Если проблема была открыта некоторое время назад**, то есть вероятность, что над ней кто-то уже работает или даже закончил работу. Не лишним будет спросить об этом;
- **Если вы открыли проблему, а затем самостоятельно её решили**, сообщите об этом и закройте вопрос. Даже документирование результата решения – это уже вклад в проект.

Открытие запроса на принятие изменений

Это делается в следующих ситуациях:

- Чтобы подтвердить простейшие поправки (опечатки, неработающая ссылка или очевидная ошибка);
- Чтобы начать работу над уже обсуждённым вкладом в проект (возможно, вы уже оговаривали его, открыв вопрос).

Как правило, такой запрос не должен представлять законченную работу. Лучше открывать его заранее, чтобы другие могли наблюдать за вашими успехами или высказывать своё мнение. Просто добавьте к теме метку «WIP» (работа в процессе). Таким образом, вы получите и возможную помощь, и сможете добавить больше коммитов в любое время.

Если проект находится на GitHub, то запрос отправляется так:

- [Сделайте форк репозитория](#) и дублируйте его на локальное хранилище. Подключите локальный репозиторий к исходному «upstream», добавив его в качестве удалённого. Проверяйте и получайте изменения из «upstream» - таким образом, вы будете в курсе происходящего и минимизируете возможность конфликтов слияния при извлечении. ([Здесь](#) более подробная информация).
- [Создайте ветку](#) для ваших правок.
- Добавьте к вашему запросу **ссылки на любые соответствующие вопросы (проблемы) или подтверждающую документацию**.
- Если ваши изменения содержат **правки в HTML/CSS**, добавьте скриншоты «до» и «после».
- **Проверяйте свои правки!** Прогоните их на всех известных тестах (если они существуют), а при необходимости – создайте новые. Независимо от наличия тестов, убедитесь, что ваши поправки не вносят хаос в существующий проект.
- **По мере возможности, соответствуйте принятому стилю проекта.** Это может значить, что вам придётся использовать отступы, точки с запятой или комментарии не так, как вы привыкли в своих репозиториях, но это упростит работу сопровождающему и облегчит поддержку ваших правок в будущем.

Если это ваш первый запрос, посмотрите видеомануал «[Make a pull request](#)» от [@kentcdodds](#). Вы можете попрактиковаться в создании запросов на принятие изменений в репозитории [First Contributions](#), созданном [@roshanjossey](#).

Что дальше?

И вот, вы сделали это! Поздравляю вас с первым опытом в области открытых проектов и надеюсь, что это лишь начало долгого и интересного пути.

После того, как вы подтвердите передачу своего вклада, возможны следующие варианты развития событий:

Никакой реакции

Мы помним, что перед началом работы нужно проверить, насколько активен проект – выше мы составляли список. Однако даже в активном проекте бывает, что участники не получают никакого ответа.

Если вы не получили ответа за неделю, можно вежливо написать в ветку с просьбой о проверке запроса. Если вы знаете человека, который может это сделать, вы можете упомянуть его (@) в этом комментарии.

Помните, что обращаться к человеку нужно **публично** – дискуссии с общественностью жизненно важны для проектов с открытым исходным кодом.

Если и после этого никто не ответит, возможно, ответа вы уже никогда не дожждётесь. Это не очень хорошее чувство, но не позволяйте ему сломить вас – такое бывает со всеми! Существует множество причин, по которым вы могли не получить долгожданный ответ, включая личные обстоятельства участников, на которые вы точно не сможете никак повлиять. Попробуйте найти другой проект или другой способ помочь этому проекту. По крайней мере, воспринимайте это как неплохую причину не тратить слишком много времени на внесение вклада, не убедившись в вовлечённости сообщества.

Кто-то попросит внести изменения

Часто вас могут попросить внести какие-либо изменения в вашу работу, причём самые разнообразные: начиная от отзывов о вашей идее и заканчивая правками в коде.

Старайтесь быть отзывчивы. Люди нашли время, чтобы уделить вам внимание, поэтому игнорировать их крайне невежливо. Если вы не знаете, как вносить изменения, изучите этот вопрос и при необходимости попросите о помощи.

Если у вас больше нет возможности уделять время этому вопросу (например, если обсуждение длится уже несколько месяцев, а ваши обстоятельства изменились), сообщите об этом сопровождающему проекта, чтобы он не ожидал от вас реакций. Возможно, кто-то будет счастлив заняться этим вопросом вместо вас.

Вам откажут в принятии вашей работы

Также может случиться так, что вашу работу попросту не примут в проект. В таком случае, мы надеемся, что вы не потратили на неё слишком много времени и сил. Если вы не понимаете причины такого отказа, можно обратиться к сопровождающему за комментариями и разъяснениями. Но – не спорьте и не превращайте этот проект во врага народа! Помните, что у вас всегда есть возможность создать свой форк или проект целиком!

Ваш вклад будет принят

В таком случае вас можно лишь поздравить с тем, что вы успешно сделали вклад в проект с открытым исходным кодом. Ура!

Вы сделали это!

Если вы только что сделали свой первый вклад в свой первый opensource-проект, или же вы в поисках способов помогать таким проектам, мы надеемся, что вдохновение вас не оставит. Даже если вам отказали, постарайтесь принять это с достоинством и поблагодарите сопровождающего – человек потратил своё время на работу с вами. В конце концов, opensource-проекты так и создаются: понемногу от каждого из нас.

2. Начало работы с проектом

Основная информация об открытом программном обеспечении и базовая подготовка к запуску своего проекта.



«Что» и «почему» opensource-проектов

Итак, вы подумываете начать работать над открытым проектом? Поздравляем! Мир оценит ваш вклад в эту сферу. Давайте разберём подробнее, что это такое – opensource-project и почему люди этим занимаются.

Что означает «открытый исходный код»?

Если проект позиционирует себя как проект с открытым исходным кодом, это означает, что **любой человек может просматривать, использовать, изменять и распространять этот проект, причём для любых целей**. Эти разрешения задокументированы в [opensource-лицензии](#).

Использование в проектах открытого исходного кода – крайне мощный шаг в его развитии, так как сама концепция opensource максимально снижает границы участия, а следовательно – позволяет быстро распространять любые идеи.

Давайте приведём пример, чтобы понять, как это работает: представьте, что у вашего друга пикничок, и вы приносите вишнёвый пирог.

- Каждый ваш друг пробует пирог (использование проекта);
- Пирог вызывает фурор! Они просят у вас его рецепт, который вы им рассказываете (изучение проекта);

- Один парень из компании – повар, и он предлагает класть в пирог поменьше сахара (внесение правок в проект);
- Другая подруга, Лиза, просит разрешения подать этот пирог на ужин (распространение проекта);

Для сравнения, рассмотрим, как же будет происходить этот же процесс в случае, когда исходный код закрыт: человек идёт в ресторан и покупает ма-а-аленький кусочек вишнёвого пирога. За каждый кусочек люди платят снова и снова, и, к тому же, рецепт пирога знает только повар в ресторане. Если же вы сможете приготовить точно такой же пирог дома и продать его как свой, ресторан подаст на вас в суд. Мне кажется, выбор довольно очевиден.

Почему люди открывают исходный код своей работы?



Одно из самых крутых преимуществ, которые я получаю от использования и работы с opensource-проектами – это появляющиеся знакомства и отношения с другими разработчиками: ведь они сталкиваются со многими из тех проблем, что и я.

@kentcdodds

Есть много причин, почему люди и компании занимаются проектами с исходным кодом:

- **Сотрудничество:** В проекты с открытым исходным кодом могут вносить правки любой человек в мире. Взять, к примеру, [Exercism](#) – платформу для выполнения упражнений по программированию, с которой работали более чем 350 участников.
- **Адаптация и перестроение проектов:** проекты с открытым исходным кодом могут использоваться любым человеком практически в любых целях. Бывает, что проекты могут использоваться просто как основа для создания чего-либо нового. [WordPress](#), например, изначально был лишь ответвлением от проекта под названием [b2](#).
- **Прозрачность.** Любой из нас может проверить проект с открытым исходным кодом на наличие ошибок или несоответствий. Данный фактор важен для стран ([Болгария](#) или [США](#)), регулируемых отраслей (банковское дело и здравоохранение), и программного обеспечения сферы безопасности, типа [Let's Encrypt](#).

Помните, что открытые проекты не ограничиваются лишь разработкой софта – сфер множество, от сбора баз данных до книг. Советуем обратить внимание на [GitHub Explore](#) – там множество идей для создания проектов.

Означает ли «opensource» «бесплатно»?

Одно из самых больших достоинств открытого исходного кода - то, что такие проекты бесплатны в использовании. Однако, это лишь «побочный продукт» множества других преимуществ.

Открытая лицензия подразумевает, что каждый сможет использовать, изменять и делиться вашим проектом в любых целях, и, следовательно, сами проекты в основном бесплатны. Если же проект стоит денег, любой человек имеет право сделать копию и использовать бесплатную версию.

В результате большинство проектов с открытым исходным кодом – бесплатны, но это не обязательное условие для открытых проектов. Существуют способы взимания платы за использование таких проектов (через двойное лицензирование или ограничение функционала), но при этом данные проекты всё равно будут соответствовать официальному определению «opensource».

Нужно ли мне запускать собственный opensource-проект?

Самый простой ответ на этот вопрос – «да». Причина так же проста: независимо от результата, запуск собственного проекта - отличный способ узнать, как вообще работает opensource.

Если вы не имеете никакого опыта в запуске проектов, и это ваша первая попытка – то, может быть, вы немного нервничаете по поводу того, что скажут люди и заметит ли вообще хоть кто-нибудь этот проект. Не нужно беспокоиться – вы не одиноки!

Работа над проектом с открытым исходным кодом очень похожа на любую другую творческую деятельность, будь то литература или живопись. И пусть представлять свою работу на всеобщее обозрение страшно, но практика и получение критики – это единственный способ стать лучше!

Если же мы до сих пор не убедили вас, то поразмышляйте над целями, которые вы преследуете.

Постановка целей

Мы расставляем приоритеты и ставим цели, чтобы понять: над чем работать, от чего отказаться и на каком моменте нам может понадобиться помощь. Начать стоит с такого вопроса: «Зачем я открываю этот проект?».

Этот вопрос может не иметь единственно верного ответа. Может быть, вы будете работать над этим проектом чтобы выполнить несколько целей; может быть и так, что для выполнения одной-единственной цели вам понадобятся несколько проектов.

Если ваша единственная цель - показать миру свою работу, то, возможно, вам даже не будут интересны вклады других участников – этот момент необходимо прописать в README. С другой стороны, если вам интересна коллективная работа над вашим проектом, то стоит потратить время на написание хорошей документации – так новичкам будет проще вливаться.



В какой-то момент я решил открыть людям доступ к UIAlertView, который я создал сам. Я изменил его, сделал более динамичным и выгрузил его на GitHub. Я впервые составлял документацию, объясняющую другим разработчикам, как использовать мой проект в своих целях. Может быть так, что никто никогда даже не попробовал его применить — это был очень простой проект – но само ощущение, что я смог что-то сделать для всех, для меня было крайне приятно.

@Mavris

По мере развития вашего проекта вашему растущему сообществу может понадобиться больше, чем просто написание кода (к примеру). Нужно будет правильно реагировать на проблемы, пересматривать определённые моменты, принимать меры по распространению проекта – всё это довольно важные задачи.

Количество времени, которое вы будете тратить на эти задачи будет зависеть от масштаба вашего проекта, и вы должны быть готовы выступить как сопроводитель проекта, чтобы решать их самостоятельно или найти кого-то, кто будет вам помогать.

Если вы являетесь частью компании, которая занимается запуском проекта, убедитесь, что у вашего проекта есть все внутренние ресурсы, необходимые для его процветания. Необходимо решить несколько вопросов: например, кто будет отвечать за сопровождение поддержку проекта после запуска, или как именно вы будете распределять задачи между сообществом.

Если вам нужен какой-либо бюджет или же персонал (для продвижения, управления или сопровождения проекта), поднимите эту тему как можно раньше.



Когда вы начинаете работать на opensource-проектом, важно убедиться, что вы учли возможности и потенциальное участие окружающего сообщества. Не бойтесь вовлекать участников, которые не связаны с вашей компанией, для работы с ключевыми аспектами проекта, особенно если эти участники – частые и полезные гости проекта.

@captainsafia

Участие в других проектах

Если ваша цель - научиться сотрудничать с другими или понять суть и принципы работы opensource-проектов, то вам стоит рассмотреть возможность участия в уже существующем проекте. Начать можно с того проекта, который вы уже используете или же такого, что импонирует вам по какой-либо причине. Принимайте любое участие – вплоть до исправления опечаток или обновления документации.

Если вы не знаете, как принимать участие в проектах, прочитайте сначала [предыдущий раздел](#).

Запуск собственного проекта с открытым исходным кодом

Нет «идеального момента» для того, чтобы раскрыть исходный код вашего проекта. Вы можете сделать это как на этапе идеи, так и на этапе работы или вообще через несколько лет работы с закрытым исходным кодом.

В двух словах – это стоит сделать, когда вам понадобится участие других людей или их отзывы.

Независимо от того, на каком этапе вы решили раскрыть проект; каждый открытый проект должен включать в себя следующую документацию:

- [Открытая лицензия](#);
- [README-файл](#);
- [Руководство по внесению вкладов](#);
- [Правила поведения](#).

Все эти документы – неплохое сопровождающее подспорье. Благодаря им сопроводителям проще координировать общение, распределять вклады участников и оберегать права каждого участника (включая и ваши собственные). Понятно, что наличие такой документации значительно увеличивает ваши шансы на положительный опыт и результат.

Если ваш проект находится на GitHub, то эти файлы, как правило, размещаются в корневом каталоге под стандартными названиями – так платформа их распознает и будет автоматически показывать читателям вашей ветки.

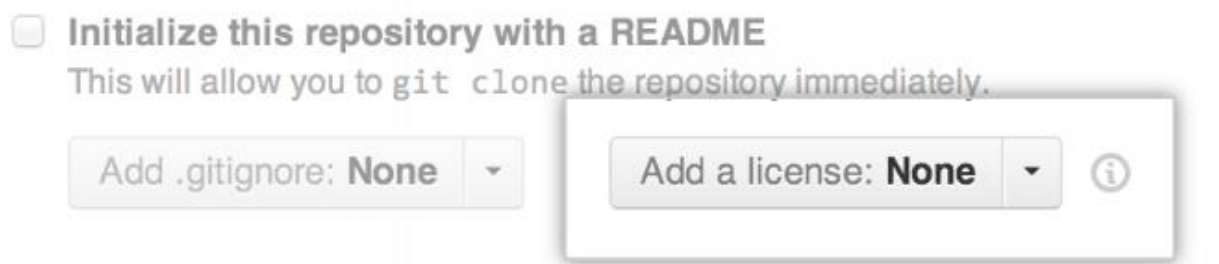
Выбор лицензии

Как уже говорилось чуть выше, открытая лицензия подразумевает, что другие люди смогут использовать, копировать, изменять и пополнять ваш проект без каких-либо последствий. Также она является отличной защитой от неприятных юридических ситуаций. **При запуске проекта с открытым исходным кодом – обязательно добавляйте лицензию.**

Юридические моменты, разумеется, никогда никому не доставляли приятных ощущений, но есть хорошая новость – вы можете скопировать и вставить существующую лицензию в свой репозиторий. Это займёт всего минуту, а проект будет защищён.

[MIT](#), [Apache 2.0](#) и [GPLv3](#) – наиболее распространённые из открытых лицензий, но ими список не ограничивается: есть и [другие варианты](#).

Когда вы создаёте новый проект на GitHub, вам предоставляется возможность выбрать лицензию. Выбор открытой лицензии сделает ваш проект также открытым.



Если у вас есть другие вопросы или проблемы, касающихся юридических аспектов управления проектами с открытым исходным кодом, мы [предоставим вам необходимую информацию](#).

Написание README

README не только объясняет, как может использоваться ваш проект: также этот файл содержит информацию о том, какое значение имеет ваш проект и почему, а также – рассказывает пользователям, что с ним можно делать.

Постарайтесь донести следующие моменты:

- Какова суть этого проекта, в чем его предназначение?
- Чем полезен этот проект?
- Как начать пользоваться проектом?
- Где получить дополнительную помощь, если понадобится?

Вы можете использовать этот файл, чтобы ответить и на другие возможные вопросы, к примеру – об распределении вкладов, целях проекта, про лицензии и так далее. Если вам пока не нужно участие других людей или же проект готов к запуску, эта информация также указывается в этом файле.



Хорошо написанная документация привлекает новых пользователей, минимизирует запросы в поддержку и мотивирует людей к участию. (...) Помните, что ваши читатели – абсолютно разные люди: в проект может прийти человек, работавший в абсолютно другой сфере и имеющий совершенно другой опыт.

@tracymakes

Иногда люди не заполняют README-файл, потому что считают проект незавершённым, либо же не желают принимать и обрабатывать вклады других участников. На самом же деле – это такие же причины, чтобы НАПИСАТЬ такой документ.

Чтобы написать полноценный README, попробуйте воспользоваться руководством @dguo «[Make README](#)» или [шаблоном](#) от @purpleBooth.

Когда вы добавляете файл README в корневой каталог, GitHub начинает его показывать на домашней странице хранилища.

Написание руководств по вкладам

Файл CONTRIBUTING даёт читателям информацию, каким образом можно принять участие в вашем проекте. К примеру, можно добавить такую информацию:

- Правила подачи отчёта об ошибке (можно использовать [готовые шаблоны](#));
- Руководство по предложению нового функционала;
- О настройке окружения и тестировании.

В дополнение к описанию технических моментов, в файле можно указать свои ожидания от сообщества:

- В какой сфере вы бы желали получать вклады от участников;
- «Путевой лист» или ваше видение проекта;
- Способы связи.

Использование тёплого, дружеского тона и указание конкретных пожеланий (например, о написании документации или создании веб-сайта) могут иметь большое значение – новички чувствуют себя желанными участниками и, соответственно, проявляют к проекту больше интереса.

Например, [Active Admin](#) начинает [эту часть документации проекта](#) с таких строк:

[Прежде всего, спасибо, что вы решили внести свой вклад в Active Admin. Именно такие люди, как вы, делают Active Admin настолько прекрасным инструментом.](#)


На ранних этапах проекта ваш файл CONTRIBUTING может быть коротким и простым. Нужно лишь объяснить, как сообщать об ошибках или проблемах с файлами, и указать, что можно сделать, чтобы принять участие в проекте (например, провести тесты).

Со временем стоит понемногу добавлять в этот файл прочие вопросы, которые вам задают пользователи – таким образом, люди будут меньше задавать одинаковые вопросы.

Чтобы получить больше информации по написанию и оформлению этого файла, ознакомьтесь с [шаблоном](#) от [@nayafia](#) или [статьёй](#) от [@mozilla](#).



Также добавьте ссылку на этот файл в README, чтобы больше людей могли его прочитать. Если вы поместите файл CONTRIBUTING в репозиторий вашего проекта, GitHub будет автоматически предоставлять его, когда участники будут открывать вопрос или оставлять запрос на принятие изменений.

Browse Issues Milestones




Please review the [guidelines for contributing to this repository](#).

Title

No one is assigned  No milestone 

Write Preview Comments are parsed with GitHub Flavored Markdown

Leave a comment 

Разработка правил поведения



Все мы сталкивались с тем, что казалось для нас оскорбительным – это мог быть и сопровождающий, который старался донести информацию, и простой пользователь, задающий какой-либо вопрос. (...) Кодекс поведения – легкодоступный документ, который указывает на то, что ваша команда очень серьезно относится к конструктивному диалогу.

@mlynch

Этот документ создан для определения основных правил поведения участников вашего проекта. Это особенно важно в том случае, когда вы запускаете проект для сообщества или какой-либо компании. Благодаря этой части документации, выстраивается здоровое и конструктивное общение, что избавляет сопровождающего от многих проблем. Для получения дополнительной информации по этому документу, мы советуем ознакомиться с [этим мануалом](#).

Помимо указания правил общения, в этом документе также можно указать различные правила для групп сообщества и сформировать систему штрафов. Как и в случае с открытыми лицензиями, существуют определённые стандарты для правил поведения, поэтому вам не придётся писать их с нуля. Самый яркий пример – [Contribution Covenant](#), используемый более чем [40,000 проектов](#), включая Kubernetes, Rails и Swift. Независимо от того, какие рамки и штрафы вы определите, вы должны быть готовы соответствовать им и применять его в отношении к сообществу.

Поместите ваши правила в файл `CODE_OF_CONDUCT`, который должен находиться в вашем корневом каталоге репозитория. Также можно дать на него ссылку в `README`.

Название и брендинг вашего проекта

Брендинг — это нечто большее, чем просто броский логотип или название. Нужно понимать, что в данном случае брендинг это то, как вы говорите о своём проекте и для кого эта речь предназначена.

Выбор правильного имени

Выберите имя, которое легко запомнить и, которое, в идеале, даст читателям представление о том, для чего предназначен ваш проект. Например:

- [Sentry](#) — это монитор приложений для создания отчётов о сбоях;
- [Thin](#) — быстрый и простой веб-сервер на Ruby.

Если вы опирались на уже существующий проект, используйте название этого проекта как префикса (например, [node-fetch](#) — это добавление `window.fetch` к Node.js).

Помните, ясность — превыше всего. Каламбуры — это, конечно, весело, но некоторые шутки могут быть непонятны или оскорбительны представителям других культур или просто другим людям. Некоторые из ваших потенциальных пользователей могут быть сотрудниками серьёзной компании: не заставляйте их краснеть, рассказывая про ваш проект на работе!

Как избежать конфликтов имён

Перед тем, как дать своему проекту название, [проверьте](#), нет ли среди существующих проектов какого-то со схожим названием (особенно таких, что используют с вами один и тот же язык или экосистему). Такие совпадения могут запутать вашу аудиторию.

Если вы хотите, чтобы ваш проект представлялся на каком-либо сайте, в Твиттере или ещё где-то, убедитесь в доступности также и этих названий. В идеале — зарезервируйте это название, даже если вы ещё не собираетесь его использовать.

Также убедитесь, что название вашего проекта не нарушает права существующих торговых марок. В этом случае, такие компании могут попросить вас закрыть проект позже или даже подать на вас в суд. Такой риск никому не нужен. Для такой проверки можно использовать [WIPO](#) — глобальную базу брендов. Если вы работаете в компании, с этим вам может [помочь](#) штатный юрист.

Последним шагом, просто поищите название в Google. Смогут ли люди легко найти ваш проект в сети? И, что не менее важно — не встречается ли в результатах поиска что-то такое, чего бы вы не пожелали видеть своим потенциальным посетителям?

То, как вы подаёте информацию (даже код), влияет на бренд!

На протяжении всей жизни вашего проекта вам придётся довольно много писать: `README`, какие-либо руководства, документация сообщества, ответы на вопросы, информационные бюллетени и так далее.

Неважно, официальная ли это документация или нет - ваш стиль письма всегда будет частью бренда вашего проекта. Подумайте, как вы хотели бы общаться со своей аудиторией и как передать этот тон.



Я старался быть участвовать в каждой ветке и демонстрировать образцовое поведение – я был доброжелателен людям, серьёзно относился к их проблемам и пытался быть полезным в целом. Через некоторое время люди стали не только задавать вопросы, но и помогать с ответами, и, к моему восторгу, они стали подавать информацию так же, как и я.

@janl

Использование тёплых, инклюзивных формулировок (таких как «они», даже если речь идёт об одном человеке) покажет вашим новым участникам, что проект настроен радушно по отношению к ним. Придерживайтесь простого языка – многие из ваших читателей могут не быть носителями английского языка.

Помимо того, как вы формулируете мысли, стиль вашего написания кода также может стать частью бренда вашего проекта. [Angular](#) и [jQuery](#) – два примера проектов с довольно строгим стилем кода и руководств.

Нет никакой необходимости писать руководство по стилю на начальных этапах работы вашего проекта – ведь вы в любой момент можете решить сменить или усовершенствовать ваш стиль. Однако нужно понимать, что ваша манера подачи информации может как привлечь, так и отпугнуть разных людей. Начальные этапы – хороший момент, чтобы это выяснить.

Контрольный список перед запуском

Вы готовы открыть ваш проект? Ниже мы приведём список, который поможет вам проверить, всё ли вы верно сделали:

Документация

- ✔ У проекта есть открытая лицензия;
- ✔ Присутствует вся документация (README, CONTRIBUTING, CODE_OF_CONDUCT);
- ✔ Название проекта легко запомнить; оно даёт некоторое представление о проекте, не противоречит ему и не нарушает никакие права;
- ✔ Все проблемы чётко организованы и размечены.

Код

- ✔ Проект использует согласованные стандарты оформления кода и определённые имена функций, методов и переменных;
- ✔ Код хорошо прокомментирован, все важные моменты отображены;
- ✔ В истории изменений, вопросах или прочих запросах отсутствуют конфиденциальные данные (например, пароли или другая непубличная информация).

Люди

Если вы работаете индивидуально:

- ✔ Вы поговорили с юристом и/или разобрались в правилах интеллектуальной собственности и политики вашей компании относительно открытых источников (если вы где-то работаете).

Если вы работаете компанией или организацией:

- ✔ Вы всё обсудили с вашим юристом;
- ✔ У вас разработан маркетинговый план для представления и продвижения проекта;
- ✔ У вас назначен ответственный за взаимодействие с сообществом (реагирование на проблемы, проверка и объединение запросов на принятие изменений);
- ✔ Как минимум два человека имеют административный доступ к проекту.

Ты сделал это!

Поздравляем с запуском вашего первого открытого проекта! Независимо от результата, такая работа — это хороший подарок для сообщества. С каждым коммитом, комментарием и запросом вы создаёте возможность учиться и развиваться и для себя, и для других людей.

3. Поиск пользователей вашего проекта

В данном разделе мы разберёмся, как «взрастить» свой проект, привлекая к нему пользователей.



Распространяйте информацию

Нет такого правила, которое обязывает вас продвигать проект с открытым исходным кодом сразу при его запуске. Многие причины, по которым люди открывают проекты, вообще не имеют ничего общего с популярностью. Вместо того, чтобы надеяться, что другие найдут и используют ваш проект, вам нужно просто рассказать миру о проделанной работе!

Сформулируйте своё послание

Прежде чем приступить к продвижению проекта, вам необходимо объяснить, для чего он создан и почему это так важно. Что делает ваш проект уникальным и интересным? Почему вы решили работать именно над этим проектом? Когда вы ответите на эти вопросы, хотя бы сами себе, вам станет проще понять всю значимость этого проекта.

Люди сначала становятся пользователями проекта, а затем и участниками по той простой причине, что этот проект каким-то образом помогает решить их проблемы. Размышляя о содержании и ценности вашего проекта, попробуйте посмотреть со стороны других людей.

Например, [@robb](#) демонстрирует использует следующие примеры кода, чтобы объяснить, почему его проект [Cartography](#) полезен сообществу:

Cartography 🗺️

Using Cartography, you can set up your Auto Layout constraints in declarative code and without any stringly typing!

Carthage compatible

build passing

In short, it allows you to replace this:

```
addConstraint(NSLayoutConstraint(
    item: button1,
    attribute: .Right,
    relatedBy: .Equal,
    toItem: button2,
    attribute: .Left,
    multiplier: 1.0,
    constant: -12.0
))
```

with this

```
constrain(button1, button2) { button1, button2 in
    button1.right == button2.left - 12
}
```



Чтобы узнать больше о том, как передать и выразить свои мысли, ознакомьтесь с упражнением «[Personas and Pathways](#)» от Mozilla.

Люди должны суметь найти и следовать вашему проекту

В идеале нужен лишь один основной URL-адрес, который и будет продвигаться и указывать людям на ваш проект. Не обязательно указывать маску или даже доменное имя, но – проекту всегда нужен фокус.

Peter Cooper & Robert Nayman

Людям будет проще найти и запомнить ваш проект, если у них будет способ это сделать. Вам нужен один простой и ясный триггер. Чтобы направить людей на ваш проект – это может быть URL GitHub, имя в Twitter или IRC-канал. Следовательно, ваше сообщество будет расти и развиваться.

Если вы пока не хотите продвигать именно проект, продвигайте личные аккаунты в Twitter или на GitHub – это позволит людям узнать, как с вами связаться или как следить за вашей работой. Если вы выступаете на встрече или мероприятии, убедитесь, в презентации есть все ваши контакты.



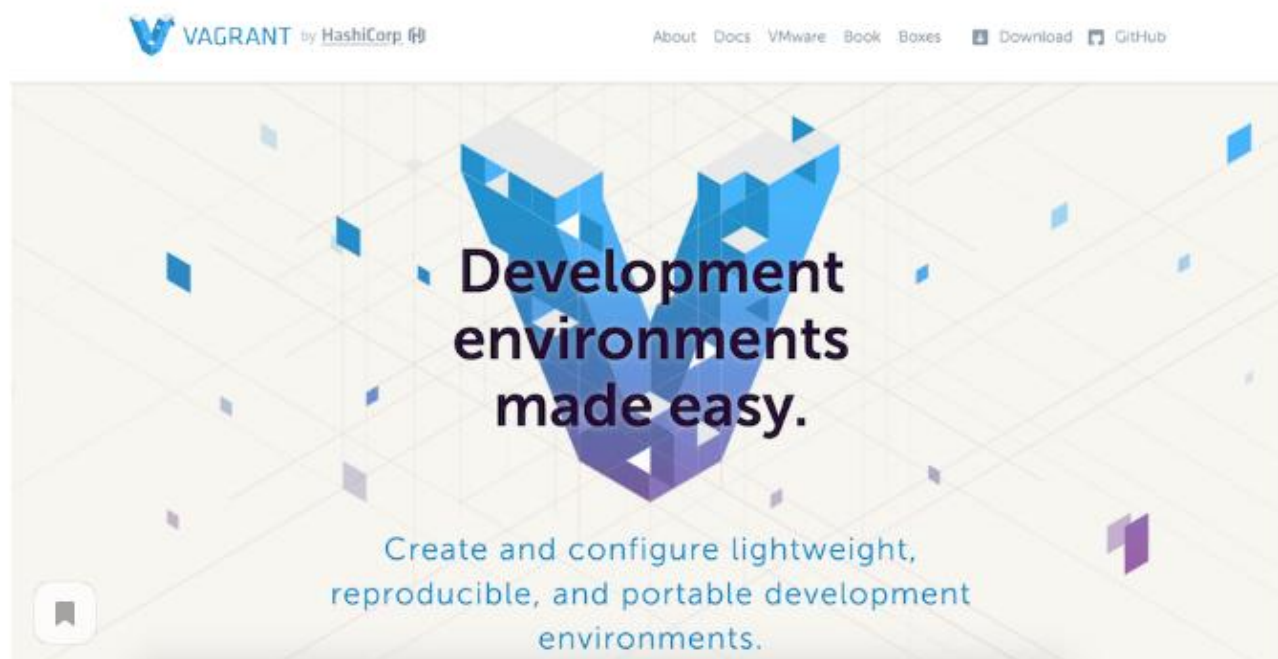
Ошибка, которую я допустил, когда начинал заниматься opensource-проектами проста – я не создал аккаунт Twitter для проекта. Twitter – отличный способ рассказать людям о проекте и информировать о его развитии.

@nathanmarz

Подумайте о создании сайта для вашего проекта – он сделает ваш проект куда более дружелюбным и удобным для просмотра, особенно в сочетании с грамотно оформленной документацией и хорошими мануалами. Наличие веб-сайта также говорит о том, что ваш проект активен, что позволит вашей аудитории чувствовать себя более комфортно при его использовании. На сайте также можно привести несколько примеров использования проекта.

[@adrianholovaty](#), один из создателей Django, сказал, что создание сайта было «безусловно, лучшим, что мы сделали для Django в начале работы».

Если ваш проект размещён на GitHub, то для создания сайта можно использовать [GitHub Pages](#). [Yeoman](#), [Vagrant](#) и [Middleman](#) - несколько [примеров](#) отличных, полноценных сайтов проекта.



Теперь, когда вы сформулировали послание для своей аудитории и настроили легкодоступные каналы связи, настало время поговорить со своими потенциальными читателями!

Онлайн-общение

Интернет – отличное средство для быстрого распространения информации. Используя онлайн-каналы, вы сможете охватить очень широкую аудиторию.

Для поиска своей аудитории, воспользуйтесь преимуществами существующих онлайн-сообществ и платформ. Если ваш проект работает над программным обеспечением, обратите внимание на [Stack Overflow](#), [Reddit](#), [Hacker News](#) или [Quora](#). Найдите каналы, участниками которых ваш проект принесёт больше всего пользы и удовольствия.



Каждая программа имеет очень специфический функционал, который будет полезен лишь небольшой части пользователей. Не нужно стараться охватить как можно больше людей – вместо этого, направьте свои усилия на те сообщества, которым будет полезно узнать о вашем проекте.

@pazdera

Вот ещё несколько способов поделиться своим проектом:

- **Узнавайте о соответствующих проектах и сообществах.** Порой напрямую продвигать свой проект вовсе не обязательно: к примеру, если он будет интересен людям, использующим Python, обратите внимание на сообщество Python-разработчиков. Когда люди узнают вас поближе, вы сможете поговорить с ними и поделиться вашей работой.
- **Найдите тех, кому поможет ваш проект.** На соответствующих форумах много людей, имеющих проблему, которую может решить ваш проект. Ответьте на их вопросы и тактично предложите свой проект в качестве решения.
- **Просите обратной связи.** Представьтесь, расскажите про свою работу аудитории, которая посчитает её актуальной и интересной. Опишите, кто, по вашему мнению, может извлечь пользу из вашего проекта. Попытайтесь закончить представление так: «Я думаю, что мой проект может помочь X, который делает Y». Общайтесь с людьми, а не просто рекламируйте свою работу.

В двух словах - сосредоточьтесь на оказании помощи другим людям. Поскольку сейчас любой может запустить онлайн-проект, создаётся мощный фон: чтобы вас выделили в нём, дайте людям понять, что вы можете, а не только чего вы хотите.

Если никто не обращает внимания или не реагирует на ваши речи, не расстраивайтесь! В основном, запуск проекта – циклический процесс, который может занять месяцы или годы. Не получили реакцию в первый раз – попробуйте другую тактику или вначале найдите способ помочь другим. Запуск и продвижение проекта требует времени и самоотдачи.

Оффлайн-общение



В реальном мире, самый популярный способ продвинуть свой проект и привлечь заинтересованную аудиторию – это различные мероприятия. Если вы же [новичок в публичных выступлениях](#), попробуйте начать с поиска небольшой встречи, связанной с областью вашего проекта.



Я очень волновалась перед посещением PyCon. Мне нужно было выступить с докладом и познакомиться с парой человек, и для того, чтобы собраться, у меня ушла целая неделя. (...) Как оказалось – это было абсолютно лишним - PyCon был потрясающим! (...) Все вокруг были невероятно дружелюбны и общительны, настолько, что я крайне редко оказывалась не вовлечённой в беседу!

@jhamrick

Если вы ещё не выступали перед публикой, то нервничать – это нормально! Просто помните, что эти люди здесь потому, что они искренне хотят услышать о вашей работе.

Когда вы будете писать свою речь, стоит сосредоточиться на том, что будет интересно для вашей аудитории – нужно понять, как извлечь из этого пользу. Будьте дружелюбны и открыты, улыбайтесь и получайте удовольствие.



Независимо от темы вашего выступления, попробуйте подать её как некую историю – так вы сможете наиболее ярко рассказать её людям.

Лена Рейнхард

Когда вы будете готовы к дальнейшим шагам по продвижению проекта, рассмотрите возможность выступить на конференции. Это поможет вам привлечь множество людей, причём иногда – со всего мира.

Искать стоит такие конференции, чьи тематики чем-то схожи с вашими проектами: например, в области того же языка или экосистемы. Перед тем, как подать заявку на участие, изучите структуру мероприятия и посетителей – так вы сможете при необходимости адаптировать своё выступление и повысить шансы на его одобрение. Простой способ разобраться в настроениях и интересах людей – послушать других участников этих конференций.



Я написал сотрудникам JSConf и попросил их предоставить мне возможность выступить на этом мероприятии. (...) Мне было очень страшно – ведь я собирался представить людям то, над чем я работал более полугода. (...) Почти всё время я просто думал «Боже мой. Что я здесь делаю?».

@ry

Создавайте репутацию

В дополнение ко всему, что мы написали выше, хотим сказать ещё одну вещь: лучший способ привлечь людей к участию в своём проекте – это участие в их проектах. Помощь новичкам, раздачи каких-либо ресурсов и вдумчивый вклад в открытые проекты других людей помогут вам приобрести положительную репутацию. Активное участие в сообществе поможет людям понять контекст вашей работы и заинтересоваться вашим проектом, а, следовательно – рассказать о

нём кому-то ещё. Развитие отношений с другими проектами может привести даже к официальным партнёрским отношениям.



Единственная причина, по которой urllib3 до сих пор является самой популярной сторонней Python-библиотекой, заключается в том, что она постоянно появляется в запросах пользователей.

@shazow

Работать над своей репутацией никогда не рано и никогда не поздно 😊 Даже если вы уже запустили свой собственный проект, продолжайте искать способы помочь другим людям. Также не существует моментального решения для быстрого формирования аудитории. Завоевание доверия и уважения окружающих требует времени, а ваша репутация будет расти (или падать) всегда – в зависимости от каждого вашего поступка.



PhantomJS был впервые выпущен в начале 2011 года. (...) Я распространял информацию простейшими способами: я писал об этом в Твиттере об этом, рассказал о возможностях проекта в блоге, упоминал о нём во время различных встреч. В 2014 году, когда проект стал более известен, я начал делать презентации.

@ariya

Держитесь! Не сдавайтесь!

До того, как люди заметят ваш проект, может пройти немало времени. Это нормально! Некоторым из самых популярных на сегодняшний день проектов потребовались годы, чтобы достичь более-менее высокого уровня. Не стоит надеяться, что проект сам внезапно станет популярным – сосредоточьтесь на формировании отношений; будьте терпеливы и продолжайте делиться своей работой с теми, кто ценит её.

4. Организация дружелюбного сообщества

Необходимые знания для создания сообщества вокруг вашего проекта, или как мотивировать людей использовать вашу разработку, участвовать в развитии и распространении проекта.



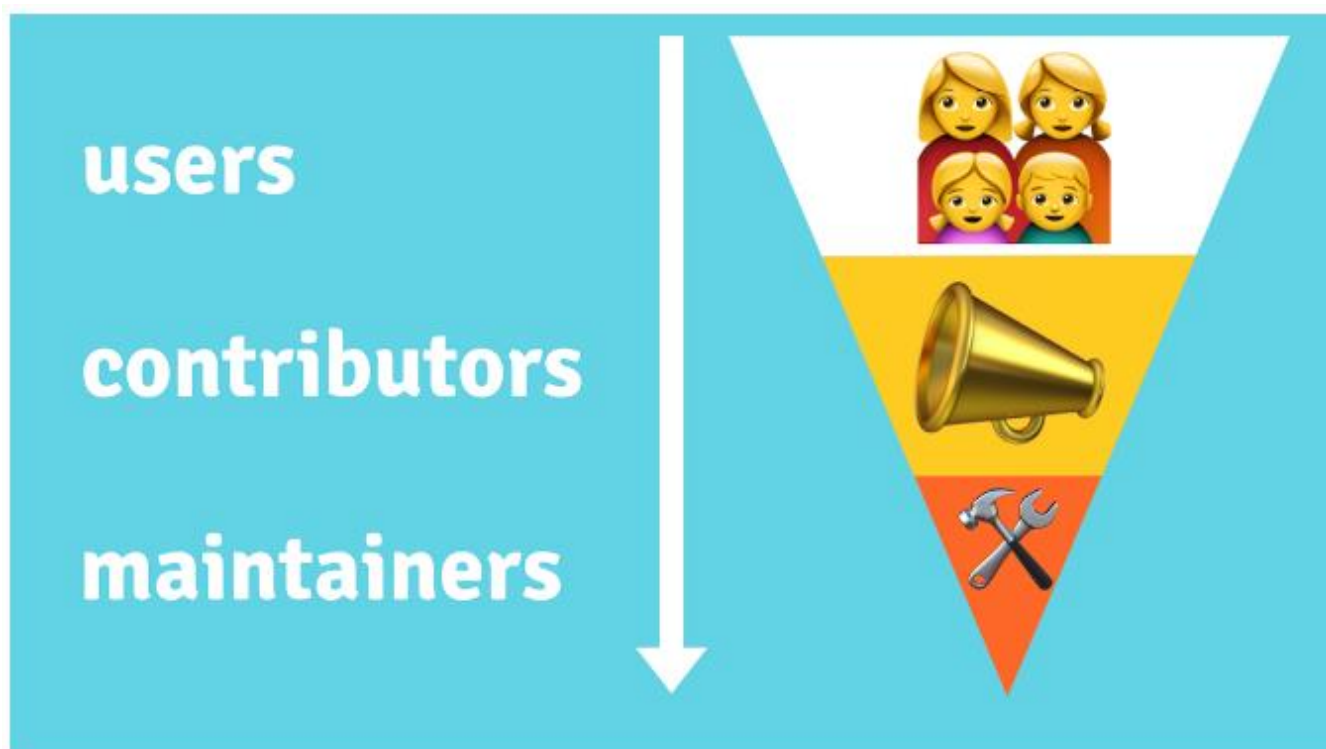
Создание условий для успешного проекта

Итак, вы запустили свой проект, рассказали про него людям, и они решили попробовать. Потрясающие! Теперь нужно понять, как сделать так, чтобы люди продолжали пользоваться проектом и захотели тоже работать над ним?

Доброжелательное сообщество — это великолепный вклад в будущее и репутацию вашего проекта. Если люди только-только начинают делать свои первые вклады, постарайтесь дать им положительные ответы (по возможности) — так они захотят продолжать работу, а новые люди увидят подобное отношение.

Заставьте людей чувствовать себя желанными

Есть один интересный способ воспринимать ваше сообщество — **@MikeMcQuaid** называет его «[воронкой участников](#)»:



При формировании сообщества, нужно понимать, что любой потенциальный пользователь (верхняя часть воронки) может в итоге пробиться к самому острию воронки – стать активным сопровождающим проекта. Вам же необходимо будет минимизировать возможные трения на каждой ступеньке этого пути. Когда люди приходят к успеху, не выворачивая себя наизнанку, они получают стимул делать ещё больше.

Начните с вашей документации:

- **Сделайте так, что ваш проект будет легко использовать.** Доступно оформленный README и понятные примеры кода смогут помочь всем потенциальным участникам.
- **Распишите процесс внесения вклада** в файле CONTRIBUTING и постоянно обновляйте его, добавляя информацию о необходимом участии.

[Опрос GitHub 2017 года](#) показал, что незавершённая либо запутанная документация проектов приносит пользователям больше всего проблем. Хорошо написанная документация же, напротив, сама по себе привлекает пользователей. Рано или поздно, кто-то откроет вопрос или разместит запрос на принятие изменений, и эти моменты нужно воспринимать как возможность затянуть этих людей в воронку.

- **Когда кто-то новый проявляет к проекту интерес, поблагодарите его!** Достаточно лишь одного негативного момента, чтобы вызвать у человека отторжение к проекту;
- **Будьте отзывчивы и будьте на связи.** Если вы не отвечаете на вопросы людей в течение месяца, то, скорее всего, они уже просто забыли о вас и вашем проекте.
- **Будьте непредубеждёнными в плане обработки вкладов,** которые вам предлагают. Многие активные участники начинают с сообщений о незначительной ошибке или небольших поправках. Как мы уже говорили, существует множество способов внести свой вклад в проект – пусть люди сами делают этот выбор.

- **Если вы не согласны с каким-то вкладом**, поблагодарите участника за идею и объясните, почему она не вписывается в рамки проекта и приведите ссылки на соответствующую документацию (если она существует).



Кому-то проще делать вклады в проекты, чем другим людям. Многие опасаются, что на них будут кричать за то, что они что-то делают не так или попросту не подходят. (...) Предоставляя участникам с низким техническим уровнем возможность вносить свой вклад (правки документации, предложения по веб-контенту и так далее), вы значительно уменьшаете эти опасения.

@mikeal

Большинство участников проектов с открытым исходным кодом – «случайные» люди, которые вносят вклады лишь изредка. Такой человек обычно не успевает полностью ознакомиться со всеми тонкостями и освоиться в этом проекте, поэтому ваша задача – максимально облегчить процесс внесения вклада для таких людей.

Поощрение других участников — это тоже своего рода вклад в будущее проекта. Когда вы позволяете людям работать над тем, что им нравится, можно быть уверенным, что они сделают всё от себя возможное.

Документируйте всё!



Вы когда-нибудь были на технической конференции, где никого не знали, а все остальные участники сбились в компании и общались, как старые друзья? (...) Теперь представьте, что вы хотите внести свой вклад в проект с открытым исходным кодом, но не понимаете, каким образом это сделать и почему все отвлечены.

@jani

Когда вы запускаете новый проект, может показаться естественным защищать конфиденциальность вашей работы. Но проекты с открытым исходным кодом процветают именно тогда, когда вы публично демонстрируете весь процесс работы. Старайтесь документировать все важные шаги – таким образом, больше людей смогут разобраться в работе и принять участие.

Ведение такого «журнала» - нечто большее чем просто запись технической документации. Каждый раз, когда вы чувствуете желание что-то записать что-то или обсудить проект с кем-то в частном порядке, подумайте, можно ли это выложить на обозрение сообществу. Не скрывайте ничего, свободно рассказывайте и про путевую карту проекта, и про виды вкладов, которые будут вам полезны, про способы их анализа и про причины принятия определенных решений.

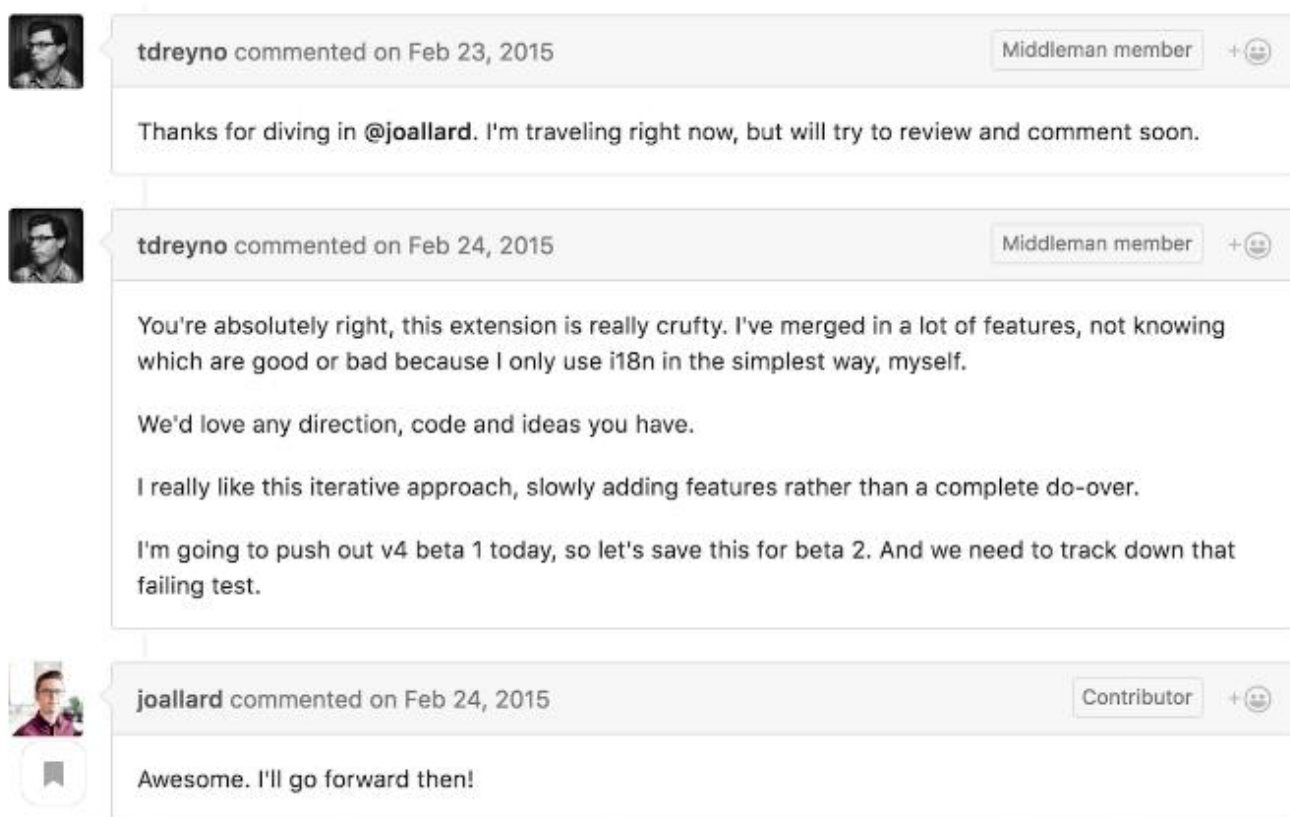
Если вы заметили, что несколько пользователей столкнулись с одной и той же проблемой, просто добавьте решение этого вопроса к README. Также рассмотрите возможность публикации своих заметок на каких-либо мероприятиях. Уровень и количество отзывов в итоге могут вас удивить.

Пополнение документации относится и к вашей выполняемой работе. Если планируется существенное обновление проекта, создайте запрос на принятие изменений и отметьте его как незавершённый (WIP). Таким образом, другие люди смогут почувствовать себя вовлечёнными в процесс.

Будьте отзывчивы

По мере продвижения вашего проекта, люди будут к вам обращаться с различными вопросами и предложениями. Старайтесь не игнорировать такие моменты – получая быстрый ответ, люди видят вашу заинтересованность и с большим энтузиазмом участвуют в проекте.

Даже если вы не можете сразу же просмотреть запрос и обработать его, покажите, что вы его заметили. К примеру, вот как **@tdreyno** отреагировал на запрос на [Middleman](#):



[Исследование](#), проведённое Mozilla, показало, что участники, которые получили отзывы в течение 48 часов, в дальнейшем делают куда больше вкладов и положительных результатов.

Также имеет смысл следить за упоминаниями вашего проекта в сети, например – на Stack Overflow, в Twitter или на Reddit. Стоит настроить систему оповещений, чтобы быть в курсе таких упоминаний.

Предоставьте сообществу возможность общаться

Для этого есть две причины:

- 1) Это полезно пользователям. Дайте людям узнать друг друга поближе – так им будет проще общаться и работать. Плюс, когда общение общедоступно и конструктивно – это неплохой способ влиться в коллектив.

- 2) Это полезно и для вас. Если не предоставить людям возможность общаться друг с другом, то они будут обращаться с каждым вопросом к вам. Поначалу это не так уж и сложно, но со временем, особенно если ваш проект станет популярным, это будет отнимать множество времени и сил. Не поддавайтесь искушению переходить в личные сообщения – предоставляйте информацию для всего сообщества.

Дать людям такую возможность – очень просто: настройте списки рассылки, создайте аккаунт Twitter, Slack или IRC-канал – и люди смогут общаться и обсуждать ваш проект. Ну и в конце концов, никто не запрещает вам использовать все эти варианты!

[Kubernetes kops](#) выделяет несколько рабочих часов каждую неделю, чтобы помочь членам сообщества:

Каждую неделю Kops выделяет время, чтобы предложить помощь сообществу и урегулировать некие вопросы. Это время отведено на работу с новичками, помощь с запросами и обсуждение новых функций.

Есть и исключения: проблемы безопасности и нарушения правил поведения должны сообщаться в индивидуальном порядке. Если вы не хотите использовать вашу личную электронную почту, создайте специальную почту под этот проект.

Развитие сообщества

Сообщества – очень мощная сила, и в зависимости как эту силу использовать, она может стать как благословением, так и проклятием. По мере роста сообщества вокруг вашего проекта нужно сделать так, что оно будет являться созидательной силой, а не разрушительной.

Не проявляйте излишней толерантности

Любой популярный проект неизбежно привлечёт людей, которые больше вредят, чем помогают. Они начинают ненужные дебаты, придираются к мелочам или запугивают новичков. Сделайте все возможное, чтобы избавиться от таких людей – если их не остановить, то они сделают некомфортным участие в проекте для других пользователей, вплоть до того, что вы начнёте терять активных участников.



Суть в том, что поддержка сообщества – для меня лично является ключевым фактором. Я никогда не смогу выполнить какую-либо работу без помощи моих коллег, дружелюбных незнакомцев или IRC-каналов для общения. (...) Не соглашайтесь на меньшее. Не потакайте засранцам.

@karissa

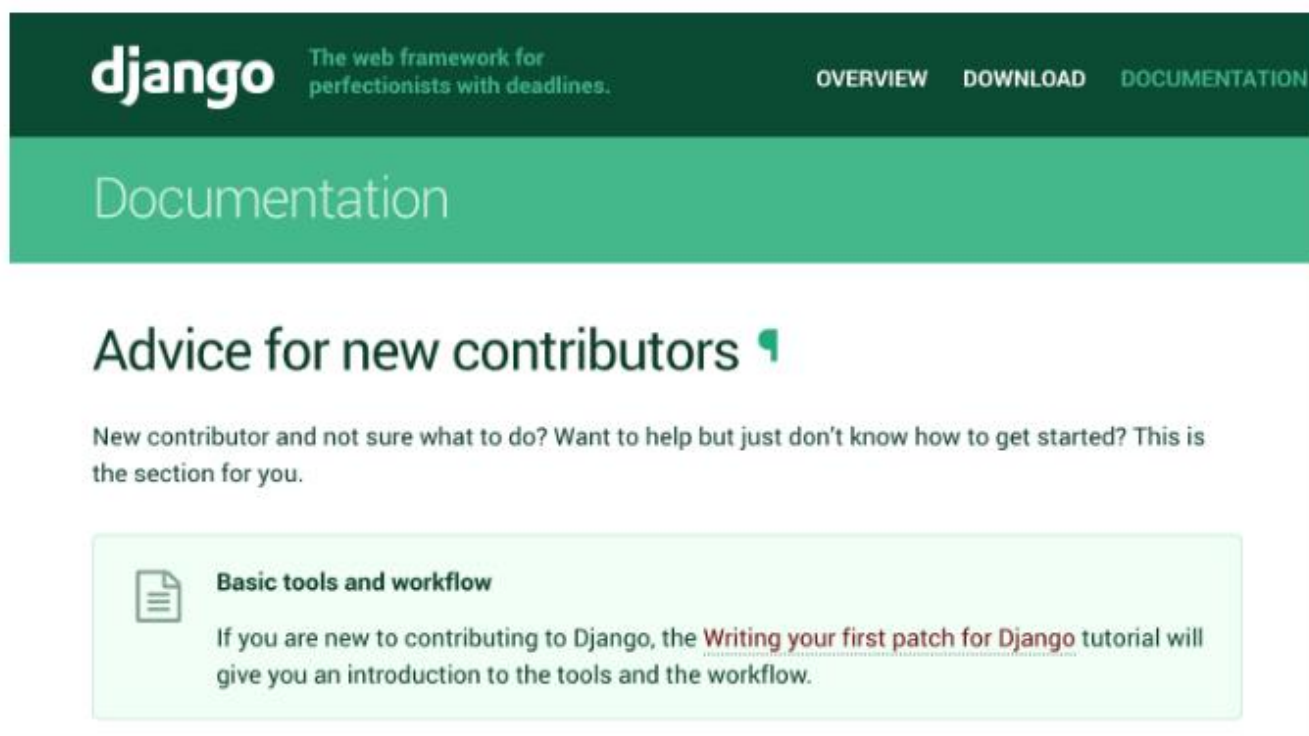
Постоянные споры по поводу тривиальных аспектов вашего проекта – отвлекающий фактор: они мешают вам и участникам сосредоточиться на важных задачах. Новички, приходящие в проект, могут вообще не пожелать работать в такой атмосфере.

Если вы увидите такое поведение, обсудите его публично. Твёрдо и тактично объясните, почему оно неприемлемо. Если это не помогает, попросите таких людей покинуть проект. [Правила поведения](#) могут стать неплохим конструктивным аргументом в такого рода разговорах.

Приветствуйте новых участников

Качество документации становится все более важным фактором развития по мере роста вашего сообщества. Случайные участники, которые могут быть вообще не знакомы с вашим проектом, получают из неё все необходимые ответы и контекст.

В файле CONTRIBUTING должна быть информация по началу работы с проектом. Для этого можно даже оформить специальный раздел. У проекта [Django](#), например, есть специальная страница, приветствующая новых участников.



Добавьте систематизацию запросов: отметьте ошибки, изучение которых будет полезно разным типам участников (к примеру, «[только для новичков](#)», «ваш первый вопрос» или «документация»). Эти [метки](#) позволяют новичкам проекта быстро сориентироваться и приступить к работе. В конце концов, используйте документацию, чтобы люди знали, что вы заинтересованы в их участии.

В большинстве случаев, вы даже ни разу не пообщаетесь с людьми, которые участвуют в вашем проекте. В документацию можно добавить материалы, полезные для таких людей. Даже пара добрых слов могут вызвать желание остаться на проекте.

Как пример мы приводим начало [одного файла](#) из документации [Rubinius](#):

[Мы хотим начать с благодарности за использование Rubinius. Этот проект – наш любимый труд, и мы ценим всех пользователей, которые находят и исправляют ошибки, улучшают производительность и помогают с документацией. Каждый ваш вклад важен для нас, поэтому – спасибо за участие! Вот несколько рекомендаций, которые могут быть полезны для вас и помогут решить ваши проблемы.](#)

Разделите управление проектом



У лидеров проекта всегда будут иметь разные мнения, как и во всех здоровых сообществах! Тем не менее, нужно уметь сделать так, чтобы победителем выходил не только самый «громкий» голос, чтобы люди слышали и менее заметные голоса, и голоса меньшинства.

@sarahsharp

Людям приятно участвовать в тех проектах, где они чувствуют себя нужными. Это не значит, что нужно полностью перевернуть видение вашего проекта или начать принимать вклады, которые вы не хотите принимать. Но чем больше вы отдаёте должное другим людям, тем больше они будут оставаться в сообществе.

Старайтесь максимально привлекать сообщество для участия в проекте. К примеру, вот несколько идей:

- **Не спешите исправлять незначительные (не критические) ошибки.** На самом деле, их стоит использовать в качестве возможности для привлечения новых участников или обучения кого-то из существующих участников. Поначалу это может показаться неестественным, но всё это с лихвой окупится со временем. К примеру, @michaeljoseph не стал исправлять проблему с [Cookiecutter](#) самостоятельно, а попросил участника оформить запрос на изменения.



prodicus commented on Mar 16

Contributor + 🗨️

The cookiecutter template mentioned in README, named `cookiecutter-python` is a dead link.



michaeljoseph commented on Mar 16

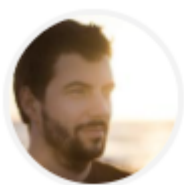
Collaborator + 🗨️

@prodicus thanks for the issue :)
Do you think you could submit a quick PR for this?

- **Создайте файл CONTRIBUTORS или AUTHORS** и перечислите всех, кто внёс свой вклад в ваш проект (такой способ использует [Sinatra](#)).
- Если у вас уже развитое сообщество, поблагодарите участников, выложив новость или написав в блоге. Два отличных примера – [This Week in Rust](#) (от Rust) и [Shoutouts](#) (от Houtie).
- **Дайте участникам доступ к коммитам.** @felixge выяснил, что это [мотивирует людей продолжать работу над патчами](#), и даже нашёл новых сопровождающих для проектов, над которыми он некоторое время не работал.
- Если ваш проект находится на GitHub, **переведите его из категории личных в категорию организации** и добавьте хотя бы одного резервного администратора. Это значительно облегчит работу с внешними сотрудниками.

Реальность такова, что в [большинстве проектов](#) существуют только один или два сопровождаителя, которые выполняют большую часть работы. Чем больше ваш проект и чем обширнее сообщество, тем легче вам будет найти помощь.

Если вы не можете с ходу найти того, кто сможет отвечать на запросы, напишите об этом – подайте людям сигнал, что вы нуждаетесь в содействии. Чем раньше вы это сделаете, тем быстрее люди смогут помочь.



В ваших интересах заинтересовать тех, кто любит и умеет делать то, чего не умеете вы лично. Вам нравится писать код, но не нравится отвечать на вопросы? Дайте это задание людям, которые умеют и любят это делать.

@gr2m

Разрешение конфликтов

На ранних стадиях проекта легко принимать важные решения: если вы хотите что-то сделать, вы просто это делаете.

По мере того, как ваш проект становится более известным, ваши решения будут интересны всё большему количеству людей. Даже если у проекта ещё не так много активных участников, но много пользователей, вы сможете найти людей, которые ищут определённых решений или выкладывают свои собственные проблемы.

По большей части, если вы создали дружелюбное, конструктивное сообщество и всегда уделяли время для документации работы, сообщество должно смочь самостоятельно найти решение. Но иногда вам придётся сталкиваться с проблемами, которые решить будет не так просто.

Установите «планку» для доброты

Когда ваше сообщество сталкивается с трудной проблемой, обстановка может накалиться. Люди могут начать злиться, расстраиваться и вступать в конфликты друг с другом или с вами. Ваша работа как сопровождаителя проекта состоит в том, чтобы не допустить обострения этих ситуаций. Даже если у вас есть собственное определённое мнение по этому вопросу, попробуйте занять непредвзятую позицию, а не вклиниваться в конфликт и продвигать свои взгляды. Если кто-то ведёт себя недоброжелательно или постоянно перебивает людей, действуйте немедленно, чтобы сохранить конструктив и адекватность дискуссий.



Сопроводителям проекта очень важно проявлять уважение к участникам. Они часто воспринимают всё очень близко.

@kennethreitz

Другие люди будут обращаться к вам за советом. Старайтесь подавать сообществу хороший пример – вам никто не запрещает выражать разочарование, огорчение или беспокойство, просто делайте это спокойно. Сохранять спокойствие нелегко, но подобное поведение хорошо влияет на ваше сообщество. Интернет скажет вам спасибо 😊

Относитесь к README как к конституции

Ваш [README](#) – это нечто больше, чем просто набор инструкций. Это отличный способ рассказать о своих целях, видении продукта и путевом листе проекта. Если люди слишком сосредоточены на обсуждении достоинств конкретной функции, то вам может быть полезно добавить в README более высокое видение вашего проекта. Фокусировка на README также обезличивает разговор, следовательно, разговор будет непредвзятым и конструктивным.

Основной фокус – на путешествии, а не на конечном пункте

Некоторые проекты используют голосование для принятия важных решений. На первый взгляд это кажется целесообразным, в том плане, что это может упростить принятие решений. Но – есть и определённые минусы: во-первых, голосование может стать явлением политического толка, когда члены сообщества будут чувствовать необходимость оказывать друг другу услугу или голосовать определенным образом; во-вторых, голосуют обычно далеко не все, причём по самым разным причинам.

Хотя иногда нам приходится идти на нарушения и всё-таки проводить голосования. Тем не менее, постарайтесь дать понять, что оно проводится не для вынесения окончательного решения, а является всего лишь «[поиском консенсуса](#)».

В рамках вышеупомянутого «поиска консенсуса» члены сообщества обсуждают основные проблемы, пока не почувствуют, что найдено коллективное адекватное решение. Когда основные проблемы решены и остаются только незначительные проблемы, сообщество движется далее. «Поиск консенсуса» признает, что сообщество не всегда в состоянии прийти к идеальному решению. Вместо этого оно отдаёт предпочтение слушанию и обсуждению.



Частичная причина того, что система голосования не работает при решении вопросов проекта Atom, заключается в том, что команда Atom не всегда будет следовать решению такого голосования. Иногда нам приходится делать выбор в пользу того, что является правильным, по нашему мнению, даже если это решение непопулярно в сообществе. (...) В чём я точно уверен, так это в том, что моя работа состоит в том, чтобы слушать сообщество.

@lee-dohm

Даже если вы, являясь сопроводителем проекта, не практикуете подобный поиск решений, важно, чтобы люди знали, что вы слушаете. Показать другим людям, что их слышать и помогать им с решением вопросов – хороший путь урегулировать любую ситуацию. Главное – стараться, чтобы ваши слова не расходились с делом.

Не торопитесь принимать решение только потому, что оно должно быть принято. Убедитесь, что вы выслушали все стороны и полученная информация доступна сообществу – только после этого можно принять обоснованное решение.

Не позволяйте разговорам уходить в сторону

Обсуждение – важный процесс, но есть огромная разница между продуктивными и непродуктивными дискуссиями.

Пока беседа идёт в правильном направлении, находится на пути к решению вопроса - поощряйте её. Но, когда становится ясно, что разговор начинает выходить за рамки (появляются личные выпады, тема ушла далеко в сторону, или же люди перемалывают одно и то же) – значит настало время его прекратить.

Если вы не препятствуете таким разговорам, то они могут негативно сказаться как на рассматриваемой проблеме, так и на состоянии сообщества в целом. Люди видят, что такое поведение допустимо вокруг этого проекта, и могут перестать вообще поднимать какие-либо темы.

Чтобы урегулировать эти моменты, попробуйте следующее:

- 1) С каждым важным замечанием, появившимся в беседе, задайте себе вопрос: «Как это приближает нас к решению?»
- 2) Если разговор начинает распыляться и теряет фокус, спросите группу: «Что нам делать дальше?»
- 3) Если разговор заходит в тупик, закройте тему и объясните, почему вы это сделали.



Направлять ветку в нужное русло и не быть при этом настойчивым – это искусство. Это непросто – уговорить людей не тратить своё время и не публиковать сообщения, если у них нет какого-то конструктивного предложения. (...) Вместо этого вы должны предложить им условия для движения вперёд, дать людям путь или подсказку, чтобы они поняли, куда идти дальше и какая дорога приведёт к желаемым результатам; но при этом вся ваша речь не должна звучать так, как будто вы диктуете им модель поведения.

@kfogel

Готовьтесь с «сражениям» с умом

Контекст очень важен. Посмотрите, кто участвует в обсуждении и какие отношения эти люди имеют с остальной частью сообщества. Разберитесь, всем ли в сообществе интересно решение этого вопроса или же это просто одинокий бузотёр? Нужно понимать, что многие люди просто молчат – не нужно сбрасывать их со счетов.

Если проблема не затрагивает большинство членов сообщества, то возможно, вам просто нужно решить её в кругу нескольких человек. Если же это уже обсуждалось, то просто дайте ссылку на предыдущие дискуссии и закройте ветку.

Определить «арбитра» сообщества

Если в сообществе царят хорошие отношения и конструктивные диалоги, самые сложные ситуации могут быть решены без особых проблем. Тем не менее, даже в продуктивной беседе может возникнуть диссонанс во мнениях по поводу того, как действовать дальше. Для этого в

сообществе должен быть арбитр – отдельный человек (к примеру, основной сопроводитель проекта) или группа людей, которые решают такие вопросы (к примеру, на основе голосования). В идеале стоит назначить арбитра и указать его (их) в файле GOVERNANCE ещё до того, как вам понадобится их помощь.

Помните, что такой способ решения проблем – «оружие последнего шанса». Разногласия и разрешение конфликтных ситуаций — это естественный процесс для роста и развития вашего сообщества. Не упускайте эти возможности и старайтесь решать проблемы сообщества везде, где это только возможно.

Сообщество – это сердце открытых проектов

Здоровые, процветающие сообщества – источник тысяч идеи и тысяч часов работы, которая в итоге становится доступна всему миру. Многие участники говорят, что именно люди всегда становятся причиной успеха – или провала - проектов с открытым исходным кодом. Научившись конструктивно использовать эту неограниченную мощь, вы сможете кому-то получить незабываемый опыт в своём проекте.

5. Советы по сопровождению проекта

В этом разделе мы разберёмся, как максимально упростить процесс сопровождения программного обеспечения с открытым исходным кодом – начиная от создания документации к процессам и работы с сообществом.



Что значит быть сопроводителем проекта?

Если вы являетесь сопроводителем крупного проекта с открытым исходным кодом, то вы должны были обратить внимание, что основная часть вашего времени уходит на то, что вы каким-то образом реагируете на проблемы.

На ранних стадиях проекта вы экспериментируете с новыми идеями и принимаете решения на основе ваших желаний. По мере роста популярности вашего проекта вы будете больше времени уделять работе с вашими пользователями и участниками.

Сопровождение проекта включает в себя довольно обширный список задач, которые часто бывают неожиданными, но от этого не менее важными для растущего проекта. Мы составили список из нескольких способов, призванных облегчить вам жизнь. Читайте и учитесь!

Документирование происходящего

Записывать все важные события, решения и запросы – важнейшее действие сопроводителя проекта: по факту, документация не только разъясняет ваши собственные идеи, но и помогает дать другим людям понять, что вам нужно или ответить на их вопросы ещё до того, как они будут заданы.

Имея на руках полноценную документацию, гораздо легче сказать «нет», когда вам что-то не подходит, следовательно, таким образом можно избежать многих щекотливых ситуаций. Помните, что любой человек может следить за вашим проектом и использовать его. Даже если вы не желаете записывать большие объёмы информации, делать маленькие заметки – это лучше, чем ничего.

Не забывайте обновлять документацию. Если у вас на данный момент нет возможности это сделать, удалите устаревшую документацию или добавьте к ней примечание – так участники будут знать, что обновления документации всегда приветствуются.

Запишите ваше видение проекта

Начните с простого – запишите цели вашего проекта. Добавьте их в README или создайте отдельный файл с именем VISION. Если у вас есть другие данные, которые могут быть полезны в этом деле (например, путевая карта проекта) – опубликуйте их тоже.

Наличие чёткого, задокументированного видения проекта помогает и вам, и участникам сосредоточиться на нужных вопросах и избежать разброса тем во вкладках участников.

К примеру, с помощью такого простого приёма @lord смог расставить приоритеты среди запросов. Как новый сопровождающий проекта, он много раз пожалел, что расплылся на запросы, не соответствующие масштабу проекта.



Первое время я возился с кучей запросов. То есть - я не мог приложить должных усилий, чтобы найти полноценное решение. Сейчас я понимаю, что вместо недолговечного решения, лучше честно сказать: «У меня сейчас нет времени на это, но я добавлю эту идею себе в список».

@lord

Сообщите людям свои ожидания

Составление свода правил может быть довольно нервным занятием. Иногда вам может показаться, что вы работаете надзирателем или на корню рубите любое веселье. Это не так: хорошо составленные и соблюдаемые правила дают сопровождающим возможность уделять больше времени проекту.

Большинство людей, которые сталкиваются с вашим проектом, ничего не знают о вас и вашей занятости. Они могут предположить, что вам платят за работу, особенно если это такой проект, которым они регулярно пользуются и от которого каким-то образом зависят. Возможно, в какой-то момент вы действительно тратили много времени на этот проект, но теперь у вас появилась другая работа или вы с головой ушли в семью – и это нормально! Людям просто нужно об этом сообщить!

Если сопровождение проекта занимает у вас не так много времени или является исключительно добровольной инициативой – чётко определите для себя, сколько времени вы готовы уделять этому делу. Нужно понимать, что эти рамки вряд ли будут совпадать с тем количеством времени, которое, по вашему мнению, нужно на самом деле уделять проекту, и уж тем более – с тем количеством времени, которого ждут от вас другие люди.

Вот несколько основных моментов, которые стоит записать, чтобы упростить себе жизнь:

- Как оценивается и принимается вклад (проводятся ли какие-либо тесты и есть ли определённый шаблон запроса)?
- В какой области принимаются вклады (может быть, вам нужна помощь только с определенной частью вашего кода)?
- Можно указать время ответа сопровождаителя (например: «Сопроводители проекта ответят вам в течение 7 дней. Если за это время вы не получили ответ, не стесняйтесь апнуть ветку»).
- Время, которое вы уделяете проекту (например: «На этот проект мы тратим около 5 часов в неделю»).

[Jekyll](#), [CocoaPods](#) и [Homebrew](#) – хорошие примеры проектов с грамотно написанными правилами для сопровождающих и участников.

Не уводите общение в приват

Не забывайте документировать все ваши взаимодействия с людьми. Везде, где только это возможно, общайтесь с людьми на виду у всех. Если кто-то попытается связаться с вами в частном порядке, чтобы обсудить запрос на принятие изменений или с просьбой помощи, вежливо направьте его на общедоступный канал связи.

Если вы встречаетесь с другими сопровождающими или принимаете важное решение наедине, также оповестите людей о результатах, пусть даже если это и будет небольшая заметка – тогда любой новичок в сообществе сможет получить доступ к той информации, которая обсуждалась годы назад.

Учитесь говорить «нет»

Итак, вы научились документировать все мало-мальски важные моменты. В идеале, каждый может прочитать эту документацию, но по факту – вам часто придётся напоминать людям, что эти записи существуют.

Однако же, если у вас есть такая «база знаний», это помогает обезличить ситуации, в которых необходимо обеспечить соблюдение правил проекта: «Ваш вклад не соответствует критериям этого проекта» звучит куда лучше, чем «Мне не нравится ваш вклад».

Как сопровождающему проекта, вы будете говорить «нет» довольно часто – если кто-то делает запрос на функции, которые не вписываются в рамки проекта, если кто-то начинает срывать конструктивный диалог, если человек выполняет ненужную работу и так далее, и так далее.

Оставайтесь доброжелательны

Чаще всего вам придётся отказывать людям в двух случаях – при рассмотрении запросов на принятие изменений и при обсуждении проблем. Как сопровождающий проекта, вы неизбежно получите те предложения, которые не захотите принимать: может быть, этот вклад изменит ожидаемый масштаб проекта, может быть – не соответствует вашему видению, а может быть – крутая идея, но реализация оставляет желать лучшего. Независимо от причины отказа, нужно отклонять такие вклады максимально тактично.

В такой момент, первой реакцией сопровождаителя может быть желание проигнорировать запрос или сделать вид, что вы его не видели. Это может обидеть или даже оскорбить участника и лишить мотивации других потенциальных участников.



Ключом к поддержке крупномасштабных проектов с открытым исходным кодом является работа с проблемами и вопросами пользователей – в этой области нужно избегать сбоев максимально ответственно. Если вы разработчик iOS, то вы знаете, как сложно работать в таком режиме: вы получаете ответ только через 2 года, и вам предлагают попробовать ещё раз, но с последней версией iOS.

@KrauseFx

Не оставляйте такой вклад открытым «просто потому, что вы чувствуете вину или хотите быть более дружелюбным». Со временем оставшиеся без ответа вопросы и запросы будут затормаживать работу над проектом и сделают её гораздо более напряжённой и пугающей. То есть – всё просто: нежелательные вклады лучше будет закрывать сразу. Если ваш проект уже страдает от накопившихся проблем, [@steveklabnik](#) составил небольшой [мануал по эффективной обработке проблем](#).

Что не менее важно, игнорирование вкладов посылает мощный негативный сигнал вашему сообществу. Люди, особенно новички, и так волнуются перед отправкой запроса, и игнорирование других запросов может вообще лишить их решимости. Даже если вы не принимаете их вклад, просто поблагодарите человека за работу и за проявленный интерес.

Итак, если вы не хотите принимать вклад:

- **Поблагодарите человека** за вклад;
- **Объясните, почему он не вписывается в рамки проекта**, и, если возможно, укажите, что можно изменить. Будьте добры, но стойте на своём;
- **Предоставьте ссылку на соответствующую документацию**, если она у вас есть. Если вы не в первый раз сталкиваетесь с подобным запросом, добавьте этот момент в свою документацию;
- **Закройте запрос.**

Чтобы ответить человеку, обычно требуется не более 1-2 предложений. Например, когда пользователь [celery](#) сообщил об ошибке, связанной с Windows, [@berkerpeksag](#) [ответил так](#):



berkerpeksag commented on Aug 13

Member



Thanks for the report. According to <http://docs.celeryproject.org/en/master/whatsnew-4.0.html#removed-features>, Windows is no longer supported by Celery 4.0.

Если мысль сказать «нет» пугает вас, знайте – вы не одиноки. Вот отличная цитата [@jessfraz](#):

Я говорил с сопровождающими нескольких различных проектов (Mesos, Kubernetes, Chromium), и все они согласны с тем, что одна из самых сложных частей нашей работы – сказать твёрдое «Нет» патчам, которые вам не нужны.

Не испытывайте чувства вины за то, что не хотите принимать чей-либо вклад. Согласно @shykes, первое правило проектов с открытым исходным кодом выглядит так: «Нет — это временно, да — это навсегда». Хотя проявлять эмпатию к энтузиазму другого человека — это хорошо, помните, что отклонение вклада — это совсем не то же самое, что отклонение человека, предложившего этот вклад.

Да и в конце концов, если вклад недостаточно хорош, вы никому не обязаны его принимать. Будьте добры и отзывчивы в общении с людьми, которые вносят свой вклад в ваш проект, но одобряйте только те изменения, которые, по вашему убеждению, сделают ваш проект лучше. Чем чаще вы говорите «нет», тем легче становится это делать. Это мы вам обещаем.

Будьте инициативны

Чтобы сократить объем нежелательных вкладов, объясните людям правила подачи запросов на вклады (в руководстве для участников). Если вы получаете слишком много подобных вкладов, попросите участников сделать какие-то пункты заранее, например:

- Заполнить анкетку или список соответствий запроса;
- Перед подачей запроса открыть проблему.

Если они не будут следовать вашим правилам, немедленно закрывайте проблему и ссылайтесь на свою документацию. Хотя такой подход может показаться жёстким, на самом деле это полезно для обеих сторон. Вам это облегчит работу, а человек сможет избежать траты множества часов на ненужную работу.



Идеальный вариант — расписать в файле CONTRIBUTING.md, что будет или не будет принято, и мотивировать людей к прочтению этой информации, прежде чем они начнут работу.

@MikeMcQuaid

Порой, получив отказ, участник может расстроиться или начать осуждать ваше такое решение. Если его поведение станет враждебным, разрядите ситуацию или, если ничего не помогает, просто удалите его из сообщества.

Практикуйте наставничество

Вполне может быть, что в вашем сообществе есть люди, которые регулярно делают вклады, не соответствующие стандартам вашего проекта. И вам, и таким людям неприятно давать и получать отказы.

Что же делать? Если вы видите, что кто-то с энтузиазмом относится к вашему проекту, но нуждается в некоторой шлифовке навыков, наберитесь терпения. Старайтесь чётко разъяснять, почему вклад не соответствует ожиданиям проекта. Попробуйте направить их на решение более лёгкой или менее двусмысленной задачи, чтобы они подобрались опыта. Если у вас есть время, вы можете какое-то время побыть их наставником или найти для этого кого-то ещё.

Используйте ваше сообщество

Вам не нужно делать все самостоятельно. Сообщество вашего проекта всегда может вам помочь: даже если у вас ещё нет активных участников, вы всегда можете обратиться к пользователям.

Поделитесь работой

Если вы ищете людей, которые могут вам помочь, обратитесь к сообществу. Если вы видите новых участников, которые часто делают вклады, предложите им более ответственную работу. Эти моменты, когда человек становится лидером, кстати, тоже следует документировать. Мотивирование других к [совместному ведению проекта](#) может значительно снизить вашу рабочую нагрузку, как верно заметила **@Imccart** в её проекте, [p5.js](#).



Я всегда говорила: «Любой может участвовать, и вам не нужно иметь большой опыт в области написания кода». У нас были люди, которые хотели прийти на мероприятие, и именно это было мне действительно интересно. Знаете, что я думала? «Появится около 40 человек, и я не смогу уделить время каждому из них...» Но люди просто собрались вместе, и это сработало. Как только один человек получал какую-либо информацию, он при необходимости делился ею со своим соседом.

@Imccart

Если вам нужно на какое-то время перестать участвовать в развитии проекта, или же вообще отойти от него – нет ничего постыдного в том, чтобы попросить кого-то другого вас заменить.

Если другие люди с энтузиазмом относятся к вашему проекту, вы можете предоставить им доступ или официально передать управление. Если кто-то сделал форк вашего проекта и активно его развивает, можно дать на этот форк ссылку. На самом деле, это очень здорово, когда много людей желают вашему проекту развития!

Ведение документации проекта **@progrium**, [Dokku](#), помогло проекту [развиваться](#) даже после того, как он покинул его:

Я написал вики-страницу с описанием того, чего я хотел и почему я этого хотел. Знаете, меня почему-то удивило, что сопровождающие начали продвигать проект именно в этом направлении! Всегда ли проект шёл по тому пути, что я наметил? Не всегда. Но он остался максимально близок к моим ожиданиям.

Позвольте людям создавать решения, которые им нужны

Если у потенциального участника видение проекта отличается от вашего, вы можете осторожно предложить им работать над собственным форком. Создание ответвления – это вовсе не плохо, ведь возможность копировать и изменять проекты – одно из самых мощных преимуществ open-source. Если люди будут работать над собственным форком, это может дать им возможность дать выход своему творчеству, не вступая в противоречие с вашим видением проекта. И это также может стать залогом к продуктивным отношениям.

То же самое относится и к пользователям, которые предлагают решение, на которое просто не хватит мощностей вашего проекта. API и определённые настройки могут помочь людям удовлетворить их собственные потребности без непосредственного изменения проекта-источника. @orta [заметила](#), что поощрение создания плагинов для CocoaPods привело к «некоторым из самых интересных идей»:

[Когда проект становится большим, почти неизбежен тот момент, когда сопровождатели становятся намного более консервативными в вопросах внедрения нового кода. Да, вы уже научились говорить «нет», но у многих людей есть законные потребности. Таким образом, самым лучшим вариантом будет трансформация вашего инструмента в полноценную платформу.](#)

Используйте роботов

Есть задачи, с которыми вам могут помочь другие люди, но есть и такие задачи, передать выполнение которых роботам будет удобнее всего. Роботы – ваши друзья: не забывайте про них, и они значительно облегчат вашу работу сопровождателем.

Проводите тесты и прочие проверки – улучшайте качество кода

Одним из наиболее значимых способов автоматизации вашего проекта является добавление тестов. Они избавляют участников от опасений, что они что-то испортят, а вам – позволят быстро анализировать и одобрять или отклонять вклады. Помните - чем более вы отзывчивы, тем более заинтересованным будет ваше сообщество.

Автоматизируйте тесты – сделайте возможность использовать их каждому участнику, и убедитесь, что они просты в использовании. Можно требовать, чтобы весь код, предлагаемый сообществом, проходил эти тесты перед подачей запроса – таким образом, вы установите минимальный стандарт качества для всех представленных материалов. [Обязательные проверки статуса](#) на GitHub гарантируют, что никакие изменения не будут добавлены без прохождения ваших тестов. Если вы добавляете тесты, то вам нужно обязательно расписать принцип их работы в файле CONTRIBUTING.



Я считаю, что тесты необходимы, что весь код, над которым работают люди, должен тестироваться. Если бы код всегда был совершенно правильным, то он не нуждался бы в изменениях – но ведь мы пишем код только тогда, когда что-то идёт не так 😊. Независимо от изменений, которые вы вносите, тесты необходимы для выявления любых случайных ошибок.

@edunham

Автоматизируйте основные задачи обслуживания

Есть и хорошие новости для сопровождающих проектов: другие люди, которые столкнулись со всеми вышеописанными сложностями и создали инструменты для их решения. Таких [инструментов](#), помогающих автоматизировать некоторые моменты сопровождения проекта, довольно много. Вот несколько примеров:

- [semantic-release](#) автоматизирует публикацию ваших релизов;
- [mention-bot](#) напоминает о запросах на принятие изменений;
- [Danger](#) автоматизирует проверку кода;
- [no-response](#) закрывает вопросы, когда автор не отвечает сопровождателям.

Также на GitHub можно найти шаблоны для [оформления проблем и запросов на принятие изменений](#): вы можете их использовать, чтобы упростить эти процессы. [@TalAter](#) написал мануал «[Choose your own Adventure](#)», в котором множество информации по оформлению запросов.

Для управления уведомлениями по почте, обратите внимание на почтовые фильтры – с их помощью можно попробовать приоритетизировать нужные письма. Если же вы хотите, чтобы всё было выполнено в каком-то определённом стиле, для этого можно использовать руководства по стилю и линтеры.

Однако учтите, наличие многих требований ко вкладам значительно повышают порог принятия. В конце концов, нам нужно чтобы эти принятые правила облегчали жизнь и вам, и участникам.

Если вы не уверены, какие инструменты следует использовать, возьмите за основу опыт других популярных проектов, особенно в чём-то схожих с вашим (например, изучите процесс принятия вкладов других модулей Node). Использование опыта схожих проектов также сделает процесс участия в вашем проекте более знакомым для людей.

Делать паузу – это нормально!

Работа с открытым исходным кодом на каком-то моменте доставляла вам радость и удовольствие; может быть так, что теперь она в какой-то мере заставляет вас чувствовать себя угнетёнными или виноватым. Бывает так, что накапливающиеся проблемы и запросы начинают внушать ощущение подавленности или даже страха.

Выгорание – это распространённая проблема среди людей, работающих в этой сфере, особенно среди сопровождающих. Ведь именно сопровождающие должны без обсуждений прикладывать все свои силы для того, чтобы проект жил и развивался.

Именно поэтому перерывы в работе так важны! Не нужно ждать, пока вы окончательно выбьетесь из сил – возьмите отпуск. [@brettcannon](#), разработчик ядра Python, решил [уйти на месяц в отпуск](#) после 14 лет добровольной работы с OSS.

Как и с любой другой работой, периодический отдых будет вас освежать, и вы будете начинать работать с новыми силами.



Поддерживая WP-CLI, я понял: чтобы приносить реальную пользу, нужно в первую очередь сделать счастливым себя. После этого я установил чёткие рамки своего участия в проекте. Для меня лучшим вариантом стали 2-5 часов в неделю, выделяемые из моего обычного графика. Таким образом, я и вовлечён в проект, и избавлен от ощущения, что это постоянная работа. Благодаря верной расстановке приоритетов, я могу регулярно добиваться успеха в том, что я считаю наиболее важным.

[@danielbachhuber](#)

Иногда бывает довольно трудно сделать перерыв в работе с проектами – создаётся ощущение, что без вас всё рухнет. Случается даже такое, что люди вас обвиняют что вы ненадолго покидаете проект.

Сделайте все возможное, чтобы найти людей, которые будут поддерживать ваших пользователей и сообщество, пока вы находитесь в отпуске. Но даже если это невозможно, всё равно сделайте перерыв, просто оповестите сообщество об этом, чтобы люди не потеряли вас.

Сделать перерыв – это не только уйти в отпуск. Если вы не хотите работать с открытым исходным кодом в выходные дни или в рабочее время, также сообщите об этом сообществу.

Главное – позаботьтесь о себе!

Сопровождение популярного проекта требует совсем другого уровня навыков, чем сопровождение проекта небольшого. Как сопроводитель, вы обретёте лидерские и личные навыки такого уровня, который доступен лишь немногим. И пусть, что это не всегда будет просто – именно такая работа, выставление жёстких рамок и одобрение только тех вкладов, что вас интересуют, помогут вам всегда оставаться счастливым, свежим и продуктивным.

6. Лидерство и управление

Основные правила принятия решений, которые помогут вам активно развивать свой проект.



Разбираемся с управлением вашего растущего проекта

Ваш проект развивается, к вам приходят новые люди, и вы полны энтузиазма продолжать эту работу. На этом этапе стоит понять, как включить постоянных и активных участников проекта в ваш рабочий процесс, причём независимо от того, будет ли это предоставление доступа к коммитам или право разрешать конфликты в сообществе. Да, у вас есть вопросы, но и у нас есть ответы.

Какие роли принимают люди открытых проектах?

Многие проекты придерживаются схожей структуры иерархии и ролей участников проекта; что же на самом деле будут значить эти роли в вашем проекте, зависит только от вас. Вот несколько примеров ролей:

- **Сопроводитель проекта;**
- **Участник проекта;**
- **Коммитер.**

В некоторых проектах, сопроводители – это единственные люди, у которых есть доступ к коммитам. В других проектах, это просто люди, которые указаны в README как сопроводители.

Сопроводитель – это совсем не обязательно тот, кто пишет код для вашего проекта. Это может быть как человек, который проделал большую работу по продвижению проекта, или же полностью оформил необходимую документацию, которая подняла доступность проекта на

новый уровень. Независимо от того, чем занимаются этим люди, основная черта сопровождаителя – чувство ответственности проект и стремление его улучшить.

Участником может быть любой человек: тот, кто комментирует проблему или запрос на принятие изменений, те люди, которые повышают ценность проекта (неважно каким путём, будь то обработка вопросов, написание кода или организация событий), или любой человек с объединённым запросом на изменение (возможно, это одно из самых узких определений участника).



Для Node.js каждый, кто появляется среди нас для того, чтобы прокомментировать проблему или добавить какой-либо код, является членом сообщества проекта. Когда мы начинаем видеть их чаще, это означает, что они перешли черту между пользователем и участником.

@Mikeal

Термин «коммиттер» может использоваться для того, чтобы обозначить различие между доступом к коммитам (это особый тип ответственности) и другими формами вклада.

Несмотря на то, что вы можете давать любые определения каждой из этих ролей, стоит рассмотреть более [широкие определения](#), чтобы поощрять больше вкладов или других форм участия. Руководящими ролями, например, можно поощрять людей, которые внесли выдающийся вклад в ваш проект, причём независимо от их технических навыков.



Вы можете знать меня как «изобретателя» Django, но на самом деле я всего лишь парень, которого наняли для работы через год после того, как Django был создан. (...) Люди думают, что я добился успеха благодаря своим навыкам программирования... Но это не так – я в лучшем случае просто неплох в этом деле.

@jacobian

Как официально оформить эти роли?

Официализация руководящих ролей даёт людям возможность проникнуться ответственностью, а другие члены сообщества будут знать, к кому обращаться за помощью.

В небольшом проекте это довольно просто – достаточно лишь добавить имена людей, выбранных на эту роль в ваш README или CONTRIBUTORS. Для более же крупного проекта, особенно если у него есть свой веб-сайт, стоит создать отдельную страницу для команды проекта и перечислить на ней лидеров. Например, у [Postgres](#) на [такой странице](#) приведены краткие профили каждого участника.

Если у вашего проекта есть активное сообщество и много участников, можно даже сформировать «основную группу» сопровождаителей или даже несколько подгруппам людей, ответственных за различные проблемные области (например, безопасность, обработка проблем или контроль поведения сообщества). Люди должны самоорганизоваться и принимать добровольное участие, выбирая роли, которые приходятся им по вкусу – не нужно заставлять их делать обратное; это принесёт куда меньше пользы в итоге.

В своё время мы дополнили основную команду несколькими «подгруппами», каждая из которых ориентирована на конкретную область проекта, например, дизайн языка или библиотек. (...) Чтобы обеспечить глобальную координацию и чёткое, согласованное видение проекта в целом, во главе каждой подгруппы находится член основной команды.

[“Rust Governance RFC”](#)

Основная группа руководителей может создать свой отдельный канал общения (например, в IRC) или регулярно встречаться для обсуждений проекта (например, в Gitter или Google Hangouts). Такие встречи можно сделать публичными, чтобы другие люди тоже могли знать, что находится на повестке дня. К примеру, в [Cucumber-ruby](#) еженедельно проводятся «[рабочие встречи](#)».

После того, как вы распределили роли руководителей проекта, не забудьте записать способ, которым люди смогут их достичь! Установите чёткие условия и определённый процесс перехода в сопроводители проекта или в члены какой-либо подгруппы проекта, и запишите эту информацию в свой файл GOVERNANCE.md.

Инструменты типа [Vossibility](#) могут помочь вам в отслеживании того, кто делает (или не делает) вклады в проект. Публичное предоставление этой информации позволяет избежать мнения о том, что сопроводители проекта – это хунта, принимающая решения в частном порядке.

Мы уже говорили об этом выше – если ваш проект находится на GitHub, то можно просто перевести проект в статус [организации](#) и добавить резервного администратора: это значительно упростит управление разрешениями и несколькими репозиториями, а также защитит будущее наследие вашего проекта.

Когда стоит предоставить кому-нибудь доступ к коммитам?

Некоторые думают, предоставлять доступ к коммитам нужно всем, кто вносит свой вклад в ваш проект – и таким образом, множество людей могут почувствовать себя важным и ответственным за этот проект.

С другой стороны, особенно в случае более крупных и обширных проектов, доступ к коммитам лучше предоставлять только тем людям, которые явно продемонстрировали свою приверженность проекту. В целом, нет единственно верного решения – вы будете принимать это решение сами!

Если ваш проект размещён на GitHub, то вы можете использовать отдельные, [защищённые ветки](#) – они дают возможность определить людей, которым можно вносить изменения в этой ветке (при определённых обстоятельствах).



Когда кто-нибудь создаёт запрос на принятие изменений, предоставьте ему доступ к коммитам. Поначалу это может казаться невероятно глупым, но следование этой стратегии позволит вам раскрыть истинную силу GitHub. (...) Когда люди получают доступ, они перестают беспокоиться о том, что их патч может остаться незамеченным и вкладывают в него гораздо больше работы.

@felixge

Типовые структуры управления для opensource-проектов

В проектах с открытым исходным кодом существует три самых распространённых типа структур управления:

- **BDFL (Вечный доброжелательный диктатор):** в этой структуре, один человек (обычно первоначальный автор проекта) имеет право окончательного решения по всем основным проектным решениям. [Python](#) – классический пример подобной структуры; также под эту категорию попадает большинство небольших проектов, так как у них всего один или два сопроводителя. Проект, курируемый организацией, также может попасть в категорию BDFL.
- **Меритократия (примечание: термин «меритократия» несёт в себе негативную коннотацию для некоторых сообществ и имеет [сложную социальную и политическую историю](#)):** при данной структуре управления официальное право принимать решения предоставляется наиболее активным участникам проекта (то есть – «за заслуги»), а решения обычно принимаются на основе консенсуса. Концепция меритократии была впервые разработана [Apache Foundation](#); [все проекты Apache](#) являются меритократичными. Вклады в таком случае принимаются только от индивидуальных участников, а не от компаний.
- **Либеральная структура вкладов:** в соответствии с этой моделью, наиболее влиятельными признаются люди, которые выполняют большую часть работы (в данном контексте идёт речь о текущей работе, а не общих «сумм» вкладов). Решения по основным проектам принимаются через поиск консенсуса (об этом мы писали чуть выше – это коллективное обсуждение основных проблем), а не голосования, и при этом рассматриваются как можно больше точек зрения. Популярные примеры проектов, использующих такую структуру – это [Node.js](#) и [Rust](#).

Какую структуру использовать – решать вам! Каждая модель имеет свои преимущества, недостатки и компромиссы. И, хотя поначалу они могут показаться совершенно разными, все три модели имеют гораздо больше общего, чем кажется с первого взгляда. Если вы решили внедрять одну из этих структур, стоит обратить внимание на эти шаблоны:

- [Шаблон структуры BDFL](#)
- [Шаблон меритократии](#)
- [Либеральная политика Node.js](#)

Нужна ли документация по управлению при запуске проекта?

Для написания подобной документации нет «идеального момента», как и для многих других моментов. Однако, это проще всего сделать тогда, когда вы лично увидите, как развивается динамика вашего сообщества. Самая лучший (и самый сложный) момент в управлении проектами с открытым исходным кодом заключается в том, что оно формируется сообществом!

Однако же стоит помнить, что определённые моменты, внесённые в документация на раннем этапе, неизбежно внесут вклад в будущий способ управления вашим проектом, поэтому начинать записи нужно чем раньше, тем лучше. К примеру, это могут быть чёткие рамки в отношении поведения или работы ваш участников.

Если вы являетесь частью компании, запускающей проект с открытым исходным кодом, перед запуском стоит провести внутреннее обсуждение того, как ваша компания планирует сопровождать проект, продвигать его и принимать важные решения. Вы также можете публично заявить, каким образом ваша компания будет (или не будет) участвовать в проекте.



Небольшие команды, работающие над проектами на Facebook, назначаются и для управления этими проектами на GitHub. Например, React управляется нативным инженером.

@caabernathy

Что если сотрудники компании начнут вносить вклады?

Успешные проекты с открытым исходным кодом используются многими людьми и компаниями, и некоторые компании могут в итоге начать получать от них доход. Например, компания может использовать код открытого проекта в качестве одного из компонентов коммерческих сервисов.

По мере того, как проект становится все более известным и широко используемым, люди, работающие в этом проекте (вы можете стать одним из них!), становятся все более востребованными и иногда получают вознаграждения за работу над этим проектом.

Важно рассматривать коммерческую деятельность как нечто абсолютно нормальное или как ещё один источник энергии для развития проекта. Разумеется, платные разработчики не должны получать особых преимуществ перед добровольцами: каждый вклад должен оцениваться только по уровню работы и техническим характеристикам. Тем не менее, люди должны чувствовать себя комфортно, как при участии в коммерческой деятельности, так и при изложении своих вариантов развития (к примеру, выступая за принятие определенного улучшения или функции).

Понятие «коммерческий» полностью совместимо с проектами с открытым исходным кодом. По факту, это просто означает, что где-то вовлечены деньги; к примеру, такое программное обеспечение используется в торговле – такие моменты становятся всё более вероятны, когда проект получает признание. (Когда программное обеспечение с открытым исходным кодом используется как часть продукта с закрытым исходным кодом, продукт остаётся «проприетарным», хотя, как и открытый исходный код, его можно использовать как в коммерческих, так и в некоммерческих целях.)

Как и все любые другие участники, коммерчески мотивированные разработчики получают влияние в проекте в зависимости от качества и количества своего вклада. Очевидно, что разработчик, которому платят за работу, может сделать больше, чем тот, кому не платят, но это нормально: оплата — это лишь один из многих возможных факторов, которые могут повлиять на то, как кто-то делает свою работу. Обсуждение вашего проекта должно быть сосредоточено именно на вкладе, а не на внешних факторах, которые позволяют людям вносить этот вклад.

Нужно ли юридическое лицо для поддержки моего проекта?

Если вы не имеете дел с коммерческой деятельностью, то для поддержки проекта вам не понадобится юридическое лицо. К примеру, если вы хотите основать коммерческое предприятие, в США вам понадобится создать C Corp или LLC. Если вы просто выполняете

контракты, связанную с вашим проектом, вы сможете принимать деньги в качестве индивидуального предпринимателя или же создать LLC.

Если вы хотите принимать пожертвования для вашего проекта с открытым исходным кодом, вы можете просто сделать на сайте или странице проекта соответствующую кнопку оплаты (например, PayPal или Stripe), и такие поступления не будут облагаться налогом (если вы являетесь некоммерческой организацией).

Многие проекты не хотят разбираться с созданием некоммерческой организации, поэтому вместо этого они находят некоммерческого финансового спонсора, который принимает пожертвования от вашего имени, обычно под процент от пожертвования. [Software Freedom Conservancy](#), [Apache Foundation](#), [Eclipse Foundation](#), [Linux Foundation](#) и [Open Collective](#) – вот примеры организаций, которые выступают в качестве финансовых спонсоров для проектов с открытым исходным кодом.



Наша цель - предоставить инфраструктуру, которую сообщества могут использовать для самообеспечения, создавая таким образом рабочее окружение, из которого каждый вкладчик, спонсор, или просто пользователь сможет извлечь конкретную выгоду.

@piamancini

Если ваш проект тесно связан с определенным языком или экосистемой, то может также существовать определённое программное обеспечение, с которым вы будете работать. Например, [Python Software Foundation](#) помогает поддерживать [PyPI](#) (менеджер пакетов Python), а [Node.js Foundation](#) помогает поддерживать [Express.js](#) (инфраструктуру, основанную на Node).

7. Получение прибыли от проекта

Получение финансовой поддержки для дальнейшего развития вашего проекта.



Почему некоторые люди ищут финансовую поддержку

Большая часть работы с открытым исходным кодом является добровольной. Мы уже знаем как это происходит: кто-то сталкивается с ошибкой в проекте, который он использует, и тут же отправляет поправку, или же человеку просто нравится возиться с проектом в свободное время.



Я искал проект-хобби, который бы занимал меня в течение недели перед Рождеством. (...) Для работы у меня был только домашний компьютер, и больше ничего. Что я сделал? Я решил написать интерпретер для языка скриптов – эта идея давно крутилась у меня в голове. (...) В качестве рабочего названия я выбрал Python.

@gvanrossum

На самом деле, есть довольно много причин, почему люди не желают получать оплату за свою работу с проектами с открытым исходным кодом.

- **Возможно, у них уже есть работа на полную ставку**, которую они любят, и это позволяет им в свободное время вносить свой вклад в открытый исходный код.

- **Им нравится думать о таких проектах как о хобби** или творчестве, и они не хотят чувствовать финансовую ответственность за работу над своими проектами.
- **Внося вклад в проекты, они получают и другие преимущества:** создают себе репутацию или нарабатывают портфолио, приобретают новые навыки или просто чувствуют себя ближе к сообществу.



Для некоторых, финансовые пожертвования действительно важны. (...) Но для нас, в мире, который весь охвачен интернетом, в быстро меняющемся мире, в мире, в котором мы живём, важно иметь возможность сказать «не сейчас, на данный момент я чувствую, что делаю что-то абсолютно другое».

@alloy

Для некоторых, особенно когда их вклады постоянные или работа над ними требует много времени, получение оплаты за участие – единственный способ, благодаря которому они могут принимать участие в проектах. Нужно понимать, что сопровождение популярных проектов может быть серьёзной обязанностью, занимающей 10 или 20 часов в неделю вместо нескольких часов в месяц.



Вы можете спросить любого сопровождающего проекта с открытым исходным кодом, и он расскажет вам о том реальном объёме работы, который приходится выполнять для управления проектом. У вас есть клиенты. Вы решаете их проблемы для них. Вы создаёте новые функции. Это, на самом деле, занимает по-настоящему много времени.

@ashedryden

Оплата работы также позволяет делать вклады людям из разных слоёв общества. Некоторые просто не могут позволить себе тратить время на проекты – ситуации бывают разные: может быть, у человека печальное финансовое положение, какие-либо долги, или, может быть, он обременён какими-то другими обязанностями. Всё это приводит к тому, что мир может никогда не получить вклад от талантливого человека просто потому, что тот не может позволить себе уделить время проектам. @ashedryden верно подметил, что это имеет определённые этические последствия, поскольку фокус работы смещён в пользу тех людей, кто уже имеет жизненные преимущества: таким образом, они могут получать дополнительные бонусы, делая вклады, а другие люди, не имеющие возможности добровольно участвовать в проектах, не получают подобных преимуществ. Всё это усиливает нехватку разнообразия в сообществах открытого исходного кода.



OSS – источник неизмеримой выгоды для технологической индустрии и, как следствие – и для всех прочих отраслей. (...) Однако, если сосредоточиться на этом смогут лишь счастливицы или одержимые, то мир потеряет огромный неиспользованный потенциал.

@isaacs

Если вы ищете финансовую поддержку, есть два пути, которые стоит рассмотреть. Вы можете получать оплату за своё время как участник проекта или же найти спонсоров для проекта.

Получение оплаты за своё время

Сегодня многим людям платят за работу с открытым исходным кодом. Самый распространённый способ начать получать «зарплату» – поговорить с вашим работодателем.

Если ваш работодатель сам использует проект, над которым вы работаете, то это станет мощным аргументом; однако, всегда есть место для творческого подхода 😊 Думайте! Возможно, он и не использует конкретно этот проект, но пользуется или увлечён Python, а сопровождение вашего проекта помогает привлекать новых Python-разработчиков. Может быть, благодаря этому, ваш босс станет выглядеть более привлекательным для разработчиков, и некоторые захотят работать на него – количество аргументом неограниченно!



Как и многие люди, занятые проектами с открытым исходным кодом, я в своё время столкнулся с бременем сопровождения проекта. Когда я только начинал заниматься проектами, я просто задерживался на работе, чтобы уделить им время, или же садился за компьютер, как только приходил домой. (...) Мне удалось обсудить со своим боссом проблемы, с которыми я столкнулся, и мы вместе выдвинули несколько идей, как мы можем коллективно решать эти задачи, учитывая наше собственное использование Babel.

@hzoo

Если у вас на данный момент нет запущенного проекта с открытым исходным кодом, над которым вы работаете, но хотите, чтобы ваша текущая работа в итоге стала opensource, попросите вашего работодателя открыть часть своего внутреннего программного обеспечения. Сейчас многие компании разрабатывают софт с открытым исходным кодом для создания своего бренда и привлечения талантливых людей. Например, **@hueniverse** обнаружил, что Walmart получает определённые выгоды, [инвестируя в проекты](#) с открытым исходным кодом. А **@jamesgpearce** пришёл к тому, что opensource-программа Facebook имеет [значительное влияние](#) на работу с кадрами:

Это тесно связано с хакерской культурой и восприятием нашей организации. Мы задали нашим сотрудникам вопрос – «Знали ли вы про opensource-программу Facebook?». Две трети ответили положительно, а половина из них сказала, что именно эта программа положительно повлияла на их решение работать на нас. Это далеко не предел, и я надеюсь, что эта тенденция будет развиваться.

Если ваша компания пойдёт по этому пути, очень важно сохранить чёткие границы между сообществом и корпоративной деятельностью. Всё же в проектах с открытым исходным кодом участвуют люди со всего мира, а это значительно больше, чем любая другая компания.



Получать оплату за свою работу с открытым исходным кодом – это редкая и замечательная возможность, но не нужно забывать про то, что вы начали этим заниматься, потому что вы страстно любите свою работу. И именно эта страсть должна являться причиной того, что компании будут вам платить.

@jessfraz

Если вы не можете убедить своего нынешнего работодателя расставить приоритеты в пользу работы с открытым исходным кодом, подумайте о поиске новой работы – в компании, которая поощряет вклады сотрудников в проекты с открытым исходным кодом. Ищите компании, которые заявляют о своей приверженности работе с открытым исходным кодом. Например:

- Некоторые компании, такие как [Netflix](#) или [PayPal](#), имеют веб-сайты, которые подчёркивают их участие в открытых источниках;
- [Zalando](#) опубликовала [политику вкладов сотрудников в открытый исходный код](#).

Проекты, создаваемые в крупной компании, такой как [Go](#) или [React](#), будут также привлекать людей для работы в них.

В зависимости от ваших личных обстоятельств, вы можете попытаться собрать деньги самостоятельно, чтобы финансировать вашу работу с открытым исходным кодом. Например:

- [@gaearon](#) финансировал свою работу над [Redux](#) через [краудфандинговую кампанию на Patreon](#);
- [@andrewgodwin](#) получил финансирование проекта по миграции схемы Django через [кампанию на Kickstarter](#);

Ну и ещё один момент – иногда проекты с открытым исходным кодом готовы оплатить решение тех или иных проблем:

- [@ConnorChristie](#) получила оплату через [bitcoin](#) за помощь [@MARKETProtocol](#) в [работе над их библиотекой javascript](#).
- [@mamiM](#) перевёл для [@MetaMask](#) необходимую информацию на японский язык после того, как проблема была [профинансирована на Bounties Network](#).

Поиск финансирования для вашего проекта

Помимо договорённостей с отдельными участниками, иногда проекты привлекают к финансированию компании или частных лиц. Финансирование может быть направлено на выплаты участникам, покрытие расходов на реализацию проекта (к примеру, плата за хостинг) или инвестирование в новый функционал. По мере того как популярность проекта растёт, поиск финансирования становится всё более простым – вот несколько распространённых вариантов.

Краудфандинговые кампании или спонсорство

Искать спонсоров легче и быстрее, если у вас уже есть сильное сообщество, хорошая репутация или если ваш проект очень популярен. Вот примеры спонсируемых проектов:

- [webpack](#) финансируется компаниями и частными лицами через [OpenCollective](#);
- [Vue](#) финансируется через [Patreon](#);
- [Ruby Together](#), некоммерческая организация, оплачивающая работу над [bundler](#), [RubyGems](#) и другими инфраструктурными проектами Ruby.

Поток дохода

В зависимости от вашего проекта, вы можете взимать плату за дополнительную поддержку, набор опций или дополнительные функции. Вот несколько примеров:

- [Sidekiq](#) предлагает платные версии с дополнительной поддержкой;
- [Travis CI](#) предлагает платные версии своего продукта;
- [Ghost](#) – некоммерческая организация с платным сервисом управления.

Некоторые популярные проекты, такие как [npm](#) и [Docker](#), привлекают [венчурный](#) капитал для поддержки развития своего бизнеса.

Грантовое финансирование

Некоторые фонды, специализирующиеся на программном обеспечении, и различные компании иногда предлагают гранты проектам с открытым исходным кодом. Бывает, что гранты выплачиваются физическим лицам (не имеющим юридического лица для ведения проекта).

- Проект [Read the Docs](#) получил грант от [Mozilla Open Source Support](#);
- Работа [OpenMRS](#) финансировалась [Stripe's Open-Source Retreat](#);
- Проект [Libraries.io](#) получил грант от [Sloan Foundation](#);
- [Python Software Foundation](#) предлагает гранты на проекты, связанные с Python.

Чтобы получить больше информации, почитайте [руководство](#) от [@nayafia](#) по получению оплаты за работу с открытым исходным кодом. Различные типы финансирования требуют разных навыков и разных подходов, поэтому сначала проанализируйте свои сильные стороны, а потом уже выбирайте подходящий вариант.

Аргументация для получения финансовой поддержки

Независимо от того, является ли ваш проект совсем молодым, или же он существует уже много лет, вам нужно хорошо подумать, как определить целевого спонсора; а чтобы убедить его финансировать ваш проект, понадобятся серьёзные и убедительные аргументы.

Чтобы быть готовым к конструктивной беседе, у вас как минимум должны быть чёткие ответы на следующие вопросы:

Влияние

Чем полезен этот проект? Почему он нравится вашим текущим пользователям и чем привлекает потенциальных пользователей? Как вы думаете, что будет с этим проектом через пять лет?

История

Попробуйте собрать доказательства того, что ваш проект на самом деле полезен и значим: это могут быть определённые метрики, отзывы, истории пользователей проекта и так далее. Хорошим аргументом будет использование вашего проекта какой-либо компанией или заслуживающими внимания людьми; отзывы известных людей также сыграют вам на руку.

Выгода спонсора

Спонсоры по своей работе часто видят различные проекты и ищут возможностей, которые им могут дать эти проекты. Почему они должны спонсировать именно ваш проект среди всех остальных? Какую пользу они получают от него?

Использование спонсорских средств

Что именно вы будете делать с предлагаемым финансированием? В этом плане лучше всего сосредоточиться на работе над основными этапами проекта и его результатах, а не на зарплате сотрудникам.

Получение средств

Есть ли у спонсора какие-либо условия выплат? К примеру, может быть так, что вам обязательно являться некоммерческой организацией или работать с некоммерческим спонсором. Или, возможно, средства должны быть предоставлены индивидуальному подрядчику, а не организации. Всё это зависит от спонсора, к которому вы обращаетесь, так что проясните этот вопрос заранее



На протяжении многих лет мы являемся ведущим ресурсом по предоставлению иконок и значков для сайтов; наше сообщество насчитывает более 20 миллионов человек, и наш контент размещается более чем на 70 миллионах сайтов, включая Whitehouse.gov. (...) С 4 версии (три года назад), веб-технологии сделали большой скачок вперёд, и, честно говоря, Font Awesome стал немного устаревшим. (...) Именно поэтому мы представляем Font Awesome 5. Мы модернизируем и изменяем CSS и полностью переделываем каждый значок. Сейчас мы говорим о лучшем дизайне и лучшей совместимости.

@davegandy

Экспериментируйте и не сдавайтесь

Получить необходимое финансирование нелегко, причём неважно, для чего – будь то проект с открытым исходным кодом, какая-либо некоммерческая организация или новое программное обеспечение. В большинстве случаев вам придётся проявить творческий подход. Если вы определитесь, как вы хотите получать деньги, соберёте необходимые данные и найдёте подходящего спонсора, то легко найдёте убедительные аргументы в пользу финансирования.

8. Кодекс сообщества

Этот раздел расскажет о том, как добиться конструктивного и адекватного поведения членов сообщества с помощью кодекса поведения сообщества.



Зачем вообще нужны правила поведения?

Кодекс поведения – это часть документации проекта, в которой прописано то, как участникам вашего проекта следует себя вести. Разработка и правильное применение этого кодекса может помочь создать дружелюбную, позитивную атмосферу в вашем сообществе.

Кодекс создаётся не только для того, чтобы членам сообщества было комфортно общаться, но и для того, чтобы вы могли избежать многих неприятных моментов. К примеру, если вы являетесь сопровождающим проекта, то со временем непродуктивные взгляды или неконструктивное общение вызовет у вас чувство усталости или неудовлетворённости своей работой.

Кодекс даёт вам возможность сформировать здоровое и конструктивное поведение вашего сообщества. Это снизит вероятность того, что вы или другие люди просто устанете от проекта, и поможет вам урегулировать те ситуации, когда вы не согласны с чьими-то действиями.

Разработка кодекса поведения

Постарайтесь написать кодекс поведения как можно раньше: в идеале – на этапе запуска проекта. Помимо чётких правил и ваших ожиданий, там должна быть следующая информация:

- Область действия кодекса (действует ли он только относительно проблем и запросов или же затрагивает и прочие действия сообщества?);
- К кому применяется кодекс (члены сообщества, сопровождающие проекта, что насчёт спонсоров?);

- Что произойдёт, если кто-то нарушит кодекс (система штрафов);
- Способы сообщить о нарушениях кодекса.

Везде, где возможно, используйте готовые решения. К примеру, [Contributor Covenant](#) — это кодекс поведения, который используется более чем 40000 проектов с открытым исходным кодом, включая Kubernetes, Rails и Swift. Ещё два хороших примера – [Django Code of Conduct](#) и [Citizen Code of Conduct](#).

Написав или скопировав кодекс, поместите файл CODE_OF_CONDUCT в корневой каталог вашего проекта и добавьте ссылки на него в файлы CONTRIBUTING и README.

Решите, как будет применяться кодекс поведения

Кодекс поведения, который не применяется (или не может быть применён), гораздо хуже, чем его отсутствие: в этом случае, люди видят, что в вашем сообществе участникам не важны и не соблюдаются прописанные правила.

[Ada Initiative](#)

Объяснить людям, как будет применяться кодекс, вы должны **до того**, как произойдёт первое нарушение. Тому есть несколько причин:

- Это показывает ваше серьёзное отношение к необходимым решениям;
- Ваше сообщество будет уверено в том, что в случае чего, проблема будет решена;
- Данными действиями вы заверите своё сообщество в том, что разбирательства будут справедливыми и прозрачными.

Также вы должны предоставить людям способ сообщать о нарушении кодекса поведения (например, адрес электронной почты) и объяснять, кто будет получать этот отчёт. Это может быть сопроводитель, группа сопроводителей или подгруппа, специально сформированная для контроля за соблюдением кодекса поведения.

Учтите, что бывает так, что нарушение совершает человек, которому приходят эти отчёты., поэтому обязательно предоставьте людям возможность сообщать о нарушениях кому-либо ещё. Например, [@ctb](#) и [@mr-c](#) (проект [khmer](#)) добавили такие [строки](#) в свой кодекс:

[О случаях оскорбительного или иного неприемлемого поведения можно сообщить по электронной почте \[khmer-project@idyll.org\]\(mailto:khmer-project@idyll.org\): письма будут отправляться только Титусу Брауну и Майклу Крузо. Чтобы сообщить о проблеме, касающейся любого из них, пожалуйста, напишите Джуду Браун Кларк, директору по вопросам разнообразия Центра изучения эволюции в действии Научно-технологического центра NSF.](#)

Также вы можете ознакомиться с [enforcement manual](#) проекта Django (хотя, может вам и не понадобится что-то настолько всеобъемлющее: всё зависит от масштаба вашего проекта).

Обеспечение соблюдения вашего кодекса поведения

В какой-то момент, несмотря на все ваши усилия, кто-то всё же нарушит этот кодекс. Существует несколько способов купирования негативного поведения при его появлении.

Соберите полную информацию о ситуации

Относитесь к мнению каждого члена сообщества так же уважительно, как и к своему. Если вы получили сообщение о том, что кто-то нарушил кодекс поведения, отнеситесь к нему серьёзно и непредвзято: такое поведение демонстрирует вашему сообществу, что вы цените их точку зрения и доверяете их суждениям.

Нарушения бывают разные – может быть, этот член сообщества ведёт себя некорректно уже не в первый раз и постоянно заставляет других чувствовать себя некомфортно, или же напротив, это может быть первое нарушение. В любом случае, это является основанием принять меры, но – уделяйте внимание контексту!

Прежде чем ринуться решать ситуацию, дайте себе время разобраться в произошедшем. Прочитайте предыдущие комментарии и обсуждения этого человека, чтобы лучше понять, кто он и почему он мог так поступить; учтите мнения других участников сообщества – не нужно судить только по своим ощущениям.

Не допустите того, что окажетесь втянутым в конфликт. Не отвлекайтесь на чужое поведение и внешние факторы. Сосредоточьтесь на том, что вам нужно и оставайтесь непредвзятым.

Stephanie Zvan

Примите соответствующие меры

После сбора и обработки информации вам нужно будет решить, что делать. На этом этапе стоит помнить, что ваша цель в качестве модератора – создать безопасную, уважительную и конструктивную атмосферу в сообществе. Подумайте не только о том, как справиться с рассматриваемой ситуацией, но и о том, как ваш ответ повлияет на дальнейшее поведение и реакцию вашего сообщества.

Когда кто-то сообщает о нарушении кодекса поведения, справиться с этим и достойно решить ситуацию становится вашей задачей. Иногда человек, сообщающий вам о нарушении, делает это с большим риском для своей карьеры, репутации или даже физической безопасности, и такое противостояние может поставить его в неудобное положение. В таком случае, вам нужно решать эту проблему непосредственно с нарушителем, если только сообщивший о нарушении не потребует публичного решения.

Существует несколько способов реагирования на нарушение кодекса поведения:

- **Дайте соответствующему лицу публичное предупреждение** и объясните, как именно его поведение отразилось на других (предпочтительно на том же канале связи, где было совершено нарушение). Такое публичное рассмотрение проблемы покажет вашему сообществу, что вы серьёзно относитесь к кодексу поведения. Будьте добры, но непреклонны.
- **Обратитесь к соответствующему человеку в частном порядке** и объясните, как его поведение отразилось на других. Такой способ можно использовать, к примеру, если ситуация связана с конфиденциальной информацией. Если вы решали проблему в частном порядке, рекомендуется вначале сообщить тому, кто отправил отчёт о нарушении о своём решении и принятых мерах.

Иногда ситуацию невозможно разрешить мирным путём - человек, о котором идёт речь, может агрессивно или даже враждебно отреагировать на ваши попытки урегулировать конфликт или просто не воспринять вас серьёзно. В этом случае вы можете рассмотреть возможность принятия более радикальных мер. Например:

- **Временно отстраните этого человека от какого-либо участия в проекте;**
- В крайних случаях – **выдайте ему вечный бан.**

Раздачу банов не следует воспринимать легкомысленно – эти меры должны приниматься только тогда, когда становится совершенно ясно, что решить ситуацию другим путём невозможно.

Ваши обязанности в качестве сопровождающего проекта

Кодекс поведения не может применяться произвольно – он един для всех. Вы, как сопровождающий проекта, несёте ответственность за соблюдение установленных правил. Это означает, что вы должны серьёзно относиться к любому сообщению о нарушении кодекса поведения – каждый сообщивший о нарушении рассчитывает на тщательное и справедливое рассмотрение его жалобы. Если вы решите, что поведение, о котором он сообщил, не является нарушением, оповестите его об этом и разъясните, почему вы не собираетесь предпринимать какие-либо действия. Дальнейший выбор будет стоять уже за ним: терпеть поведение, которое ему не по душе или же прекратить участие в проекте.

Сообщение о поведении, которое *технически* не нарушает кодекс поведения, всё равно может указывать на наличие проблем в сообществе, и вам следует уделить внимание этим потенциальным проблемам. Может, вам придётся пересмотреть кодекс поведения, чтобы внести в него некоторые уточнения или поговорить с человеком, о поведении которого идёт речь (сообщить ему, что, хотя он и не нарушил кодекс поведения, его действия заставляют людей чувствовать себя некомфортно).

В конце концов, всё зависит от вас: как сопровождающий проекта, вы сами устанавливаете и следите за выполнением стандартов приемлемого поведения. У вас есть потрясающая возможность формировать общественные ценности проекта, и ваше сообщество рассчитывает, что вы будете это делать справедливо и беспристрастно.

Поощряйте поведение, которое вы хотите видеть вокруг себя

Когда проект начинает казаться враждебным или просто недоброжелательным, даже если это чувство возникает из-за всего лишь одного человека, вы рискуете потерять гораздо больше участников, чем в случае отстранения этого человека (или людей) от проекта. Не всегда легко принять правильное решение, но создание благоприятной среды в сообществе поможет росту вашего сообщества.

9. Метрики проектов

Руководство по грамотному отслеживанию метрик проекта – это поможет вам принять верные и обоснованные решения по развитию вашего проекта.



Зачем вообще что-то измерять?

Полученные в результате анализа метрик данные, при правильном использовании, могут помочь вам, как сопроводителю проекта, принять правильные решения в отношении вашего проекта.

С этими данными у вас есть возможность:

- Понять, как пользователи воспринимают новый функционал;
- Узнать, откуда приходят новые пользователи;
- Решить, продолжать ли поддержку какой-либо функции;
- Увидеть в цифрах популярность вашего проекта;
- Понять, как используется ваш проект;
- Получать средства через спонсорство или гранты.

Например, сопроводители проекта [Homebrew](#) считают, что Google Analytics помогает им расставить верные приоритеты в работе:

[Homebrew распространяется на бесплатной основе и полностью контролируется добровольцами в их свободное время. Как результат такой политики – мы не имеем достаточно ресурсов для подробного изучения мнений пользователей Homebrew, которые важно знать для того, чтобы решить, как лучше разрабатывать новый функционал и как расставить приоритеты работы на текущий момент. Анонимная совокупная аналитика](#)

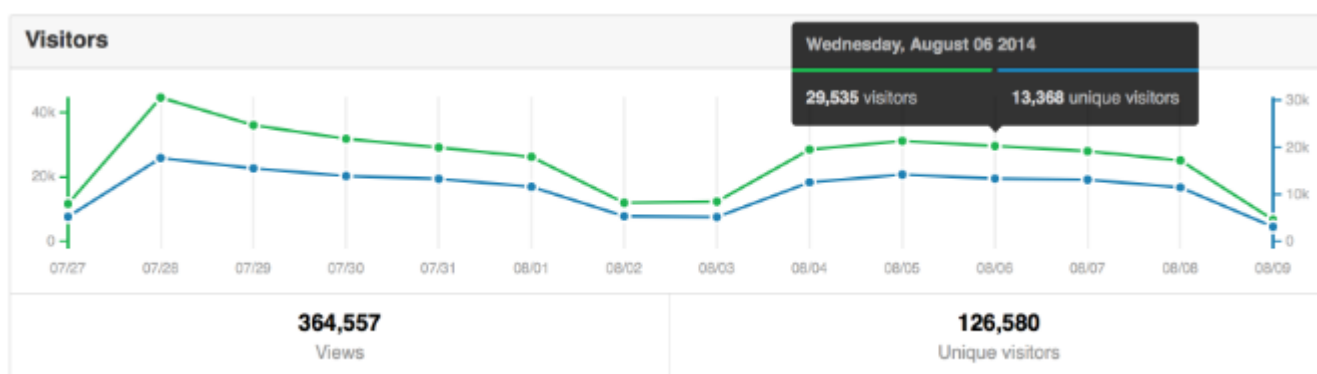
пользователей позволяет нам определять, чем заняться в первую очередь – это зависит от того, как, где и когда люди используют Homebrew.

Популярность – это ещё не всё. Мы уже говорили о том, что у каждого своя причина работы в открытых проектах: если ваша цель, как сопровождающего проекта, состоит в том, чтобы поделиться своей работой, похвастаться своими идеями или просто получить удовольствие от процесса, то показатели могут не иметь для вас никакого значения.

Если же вы заинтересованы в анализе и понимании вашего проекта на более глубоком уровне, то вам будет интересно узнать побольше о метриках проекта.

Как люди находят ваш проект?

Согласитесь, что перед тем, как кто-либо сможет использовать ваш проект или внести в него свой вклад, ему нужно каким-то образом узнать об этом проекте. Как же люди находят ваш проект?



Если ваш проект размещён на GitHub, то эту [информацию можно получить](#), зайдя со страницы проекта на страницу «Insights», а затем на «Traffic». Вот что вы увидите:

- **Общее количество просмотров:** сообщает, сколько раз был просмотрен ваш проект;
- **Количество уникальных посетителей:** число людей, просмотревших ваш проект;
- **Ссылки:** сообщает, откуда посетители перешли на вашу страницу. Этот показатель может помочь вам определить охват аудитории и эффективность рекламных кампаний;
- **Популярный контент:** самые посещаемые страницы проекта (рассчитывается на основе просмотров страниц и уникальности посетителей).

[GitHub stars](#) также может помочь вам в определении базового показателя популярности проекта. Хотя GitHub stars и не связан напрямую с загрузками и использованием проекта, эта метрика может сообщить вам, скольким людям интересна ваша работа.

Также для отслеживания можно использовать инструменты Google PageRank, счётчики реферального трафика с веб-сайта проекта или других проектов и сайтов.

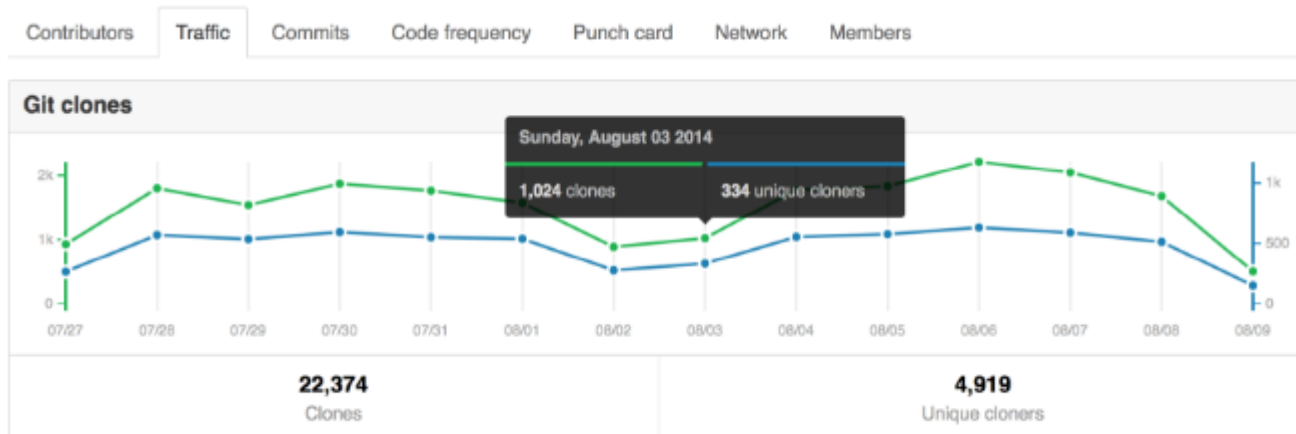
Использование проекта

Допустим, люди нашли ваш проект в этом диком и сумасшедшем пространстве, которое мы называем Интернетом. В идеале, когда они наталкиваются на ваш проект, они предпримут какие-либо действия. Вот и второй интересующий нас вопрос: используют ли люди этот проект?

Если для распространения проекта вы используете менеджер пакетов (к примеру, npm или RubyGems.org), то вы сможете отслеживать загрузки вашего проекта.

Каждый менеджер пакетов использовать своё понимание «загрузок», и то, что человек загрузил ваш проект, не обязательно значит, что он его установит или будет использовать, но это уже определённая статистика. Для отслеживания статистики использования многих популярных менеджеров пакетов, попробуйте Libraries.io.

Если ваш проект размещён на GitHub, то на странице «Traffic» есть [график клонирования](#) – он покажет, сколько раз за день ваш проект был клонирован, учитывая уникальность посетителей.



Если количество людей, использующих ваш проект гораздо ниже количества людей, которые его просматривают, вам нужно думать в 2 направлениях:

- Либо ваш проект не конвертирует вашу аудиторию должным образом;
- Либо же вы просто привлекаете не ту аудиторию.

Например, если ссылка или информация о вашем проекте попадает на первую страницу портала Hacker News, вы, вероятно, увидите резкий всплеск переходов, но, при этом, низкий коэффициент конверсии. Однако если ваш проект по Ruby представлен на конференции Ruby, вы получите высокую конверсию, так как это - целевая аудитория.

Постарайтесь понять, откуда привлекается ваша аудитория, и попросите людей оставлять отзывы на странице вашего проекта – так вы узнаете, с какой из этих двух проблем вы столкнулись.

Как только вы убедитесь, что люди используют ваш проект, вы можете попытаться выяснить, как именно они его используют. Делают ли они форки, добавляют ли функционал, используют ли ваш проект для науки или бизнеса?

Сохранение пользователей

Итак, люди нашли ваш проект и используют его. Следующий вопрос: делают ли они вклады?

Никогда не рано начать думать о том, вкладываются ли люди в ваш проект. Без помощи и участия других людей вы рискуете оказаться в неприятной ситуации, когда ваш проект уже стал популярен, но времени сопровождаителей не хватает на его полноценную поддержку.

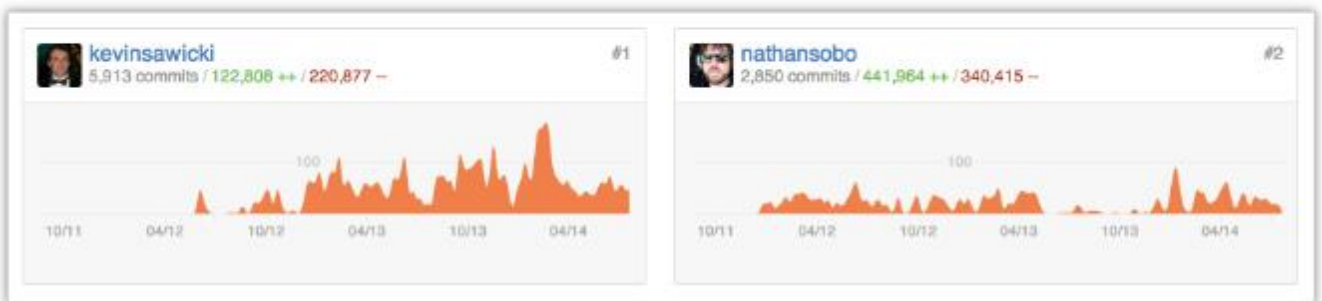
Очень важно обеспечить приток новых участников, так как активные участники рано или поздно перейдут к другим делам и проектам. Примеры метрик сообщества, которые вы можете отслеживать, включают:

- **Общее количество участников и количество коммитов на каждого участника.** Так вы сможете узнать, сколько участников у вашего проекта на данный момент, и кто из них более или менее активен. На GitHub эта информация расположена на странице «Insights» -> «Contributors». На данный момент этот график учитывает только тех участников, которые оставляют коммиты на ветке репозитория по умолчанию.

Aug 14, 2011 - Jul 23, 2014

Contributions to master, excluding merge commits

Contributions: **Commits** ▾



- **Новички, случайные и регулярные участники:** эта метрика покажет вам, приходят ли на ваш проект новички и возвращаются ли они во второй раз. Случайные участники – это участники с небольшим количеством коммитов, независимо от того, один ли это коммит, менее пяти коммитов или что-то ещё – это зависит от ваших настроек. Без новых участников сообщество может перейти в состояние стагнации.
- **Количество открытых проблем и открытых запросов на принятие изменений.** Если эти цифры слишком велики, вам может понадобиться помощь в обработке проблем и проверке кода.
- **Общее количество проблем и запросов на принятие изменений.** Открываемые проблемы означают, что кто-то желает помочь вашему проекту. Если это число со временем увеличивается, это говорит о том, что люди заинтересованы в вашем проекте.
- **Типы вкладов:** тут отображается, какие именно вклады делают люди: например, коммиты, исправления опечаток/ошибок или комментирование проблем.



Открытый исходный код – это больше, чем просто код. Успешные проекты с открытым исходным кодом – это совокупность кода, документации и всех обсуждений возможных изменений.

@arfon

Деятельность сопровождаителей

Наконец, вам нужно убедиться, что сопровождаители вашего проекта смогут справиться с получаемым объёмом работы. Последний вопрос: отвечаю ли я сообществу?

Сопроводители, игнорирующие вопросы и запросы сообщества – слабое место любого проекта с открытым исходным кодом. Если кто-то делает вклад, но не получает от сопровождающего никакой реакции, человек может просто разочароваться и покинуть проект.

[Исследование](#) Mozilla явно демонстрирует, что именно полноценная реакция – самый важный фактор при принятии участником решения, работать дальше с этим проектом или нет.

Посмотрите и разберитесь, сколько времени вам (или другому сопровождающему) требуется для реакции на разные вклады, будь то проблема или запрос на принятие изменений. Учтите, что ответ – это не обязательно решение проблемы или принятие каких-то мер. Это может и простое сообщение типа «Спасибо за ваш вклад! Я рассмотрю его в течение следующей недели.»

Вы также можете отследить время, необходимое для перехода между ступенями процесса вклада, например:

- Среднее время, сколько проблема остаётся открытой;
- Закрываются ли проблемы оформлением запроса на принятие изменений;
- Закрываются ли устаревшие проблемы;
- Среднее время, требуемое на одобрение запроса на принятие изменений.

Используйте метрики, чтобы узнать больше о людях

Использование, понимание и анализ метрик поможет вам создать активный, растущий проект с открытым исходным кодом. Даже если вы следите за всеми метриками через панель управления, используйте описанные выше способы, чтобы сосредоточить своё внимание на тех действиях, которые помогут вашему проекту процветать.

10. Юридическая сторона открытого программного обеспечения

В этом разделе вы найдёте всю важную информацию о юридическом подтексте проектов с открытым исходным кодом – всё что вы хотели узнать и ещё кое-что 😊



Разбираемся в правовых моментах

Предоставление своей работы миру может быть захватывающим и полезным опытом. Но также – это множество юридических моментов, о которых вы не знали, и о которых следует беспокоиться. К счастью, вам не нужно начинать с нуля – мы постараемся дать вам всю возможную информацию. Внимание – прежде чем вы начнёте изучать предоставленный материал, обязательно прочитайте наш [отказ от ответственности](#).

Почему людей так волнует юридическая сторона проектов?

Когда вы создаёте какое-либо творческое произведение (к примеру, тексты, рисунки или код), по умолчанию вы будете иметь исключительное авторское право на это произведение. То есть, по закону вы можете сами выставлять рамки того, что другие люди могут делать с вашей работой.

В общем и целом, это означает, что никто другой не может использовать, копировать, распространять или изменять вашу работу, не подвергаясь рискам вплоть до судебных разбирательств.

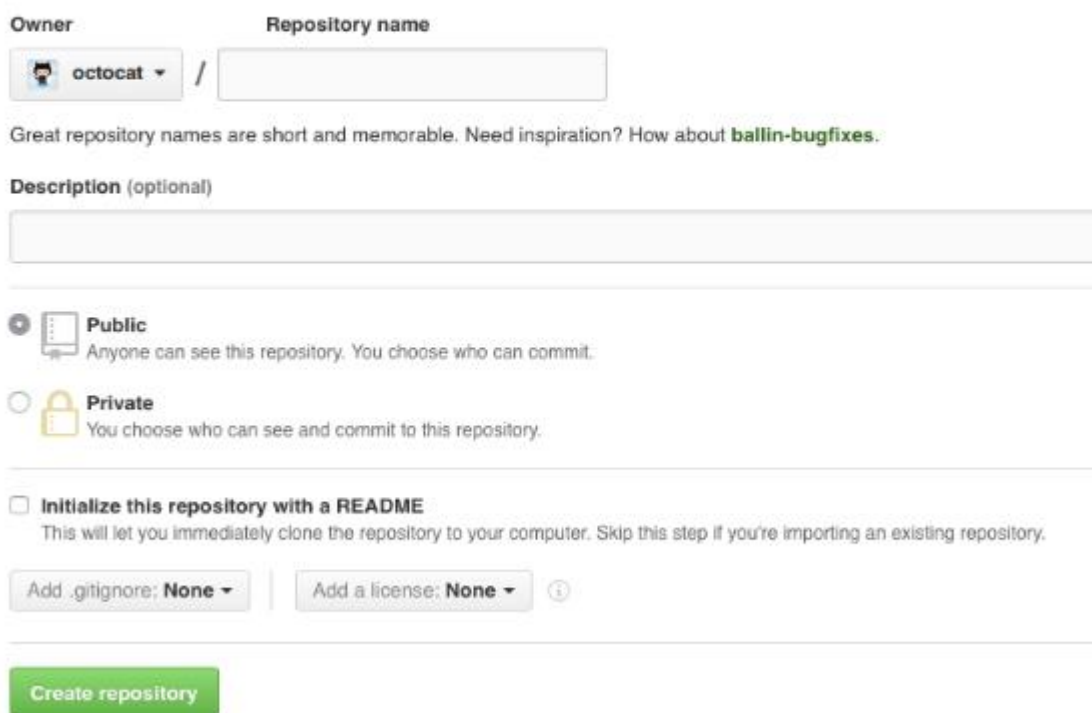
Однако проекты с открытым исходным кодом – отдельный разговор, так как автор подразумевает, что другие люди будут использовать, изменять и делиться его работой. Но, поскольку авторские права никуда не деваются, вам нужна лицензия, в которой прямо указаны эти разрешения.

Если вы не применяете открытую лицензию к вашему проекту, каждый, кто участвует в вашем проекте, также становится эксклюзивным правообладателем своей работы. Это означает, что никто не может использовать, копировать, распространять или изменять материалы проекта - и понятие «никто» также включает вас.

Ну и в конце концов, ваш проект может зависеть от каких-либо лицензионных требований, о которых можете даже не знать. Нужно понимать, что сообщество вашего проекта или политики вашего работодателя могут обязать вас использовать определённую открытую лицензию – мы рассмотрим подобные ситуации ниже.

Являются ли публичные проекты GitHub open-source-проектами?

Когда вы создаёте новый проект на GitHub, у вас есть возможность сделать репозиторий **частным** или **общедоступным**.



The screenshot shows the GitHub repository creation interface. At the top, there are two fields: 'Owner' with a dropdown menu showing 'octocat' and a 'Repository name' text input field. Below these is a hint: 'Great repository names are short and memorable. Need inspiration? How about **ballin-bugfixes**.' There is a 'Description (optional)' text area. Below the description, there are two radio button options: 'Public' (selected) with the description 'Anyone can see this repository. You choose who can commit.' and 'Private' with the description 'You choose who can see and commit to this repository.' Below the radio buttons is a checkbox labeled 'Initialize this repository with a README' with the subtext 'This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.' At the bottom, there are two dropdown menus: 'Add .gitignore: None' and 'Add a license: None', followed by a green 'Create repository' button.

Публикация вашего проекта на GitHub — это не то же самое, что лицензирование вашего проекта. На публичные проекты распространяется действие [Условий предоставления услуг GitHub](#) – они позволяют другим пользователям просматривать и создавать форки вашего проекта, но на этом список разрешений заканчивается.

Если вы хотите, чтобы другие люди использовали, распространяли, модифицировали или вносили свой вклад в ваш проект, вам необходимо добавить в проект открытую лицензию. Смысл в том, что без этой лицензии никто не сможет на законных основаниях использовать какую-либо часть вашего проекта GitHub, даже если он является публичным.

Просто дайте мне всё для защиты проекта без [TL;DR](#)

На сегодняшний день открытые лицензии стандартизированы и просты в использовании: всё настолько просто, что вы можете просто скопировать и вставить готовую лицензию прямо в ваш проект.

[MIT](#), [Apache 2.0](#) и [GPLv3](#) – наиболее популярные лицензии, но есть и другие варианты; полный текст этих лицензий и инструкции по их использованию можно найти на [choosealicense.com](#).

GitHub [предлагает вам добавить лицензию](#) каждый раз, когда вы создаёте новый проект.



Стандартная лицензия служит для людей без юридической подготовки своего рода инструкцией, что можно и что нельзя делать с программным обеспечением. За исключением случаев, когда это абсолютно необходимо, старайтесь избегать изменённых или нестандартных терминов в своей лицензии – они станут лишь препятствием для работы с проектом.

@benbalter

Какая лицензия подходит для моего проекта?

Если вы начинаете проект «с чистого листа», то лучше всего выбрать лицензию [MIT](#). Она короткая, очень простая для понимания и позволяет любому человеку делать что угодно с вашей работой, если у него есть копия лицензии и указание ваших авторских прав. Если это когда-либо понадобится, можно будет выпустить проект под другой лицензией.

Если же у вас другой случай, то выбор подходящей лицензии для вашего проекта зависит от ваших целей. Ваш проект, скорее всего, имеет (или будет иметь) **зависимости**. Например, если вы запускаете проект на базе Node.js, вы, вероятно, будете использовать библиотеки из диспетчера пакетов Node (npm). Каждая библиотек, от которой зависит ваш проект имеет собственную лицензию; если эти лицензии дают разрешение на использование, изменение и распространение без каких-либо условий для, вы можете использовать любую лицензию по своему усмотрению. Обычные в таких лицензиях библиотек прописываются MIT, Apache 2.0, ISC и BSD. С другой стороны, если какая-либо из лицензий ваших зависимостей имеет чёткие указания, тогда вашему проекту придётся использовать указанную. Обычно в таком случае используются GPLv2, GPLv3 и AGPLv3.

Вы также можете обратить внимание на **сообщества**, с которыми вы надеетесь иметь дело:

- **Хотите ли вы, чтобы ваш проект использовался в качестве зависимости другими проектами?** Тогда, вероятно, лучше всего будет использовать лицензию, самую популярную в соответствующем сообществе. К примеру, для [библиотек npm](#) это [MIT-лицензия](#).
- **Хотите, чтобы ваш проект был привлекательным для крупного бизнеса?** В таком случае вам лучше всего выбрать явную патентную лицензию, например, [Apache 2.0](#).
- **Ваш проект будет работать с авторами, не желающими, чтобы их вклады использовались в программном обеспечении с закрытым исходным кодом?** Тогда ваш выбор – [GPLv3](#) или [AGPLv3](#) (в том случае, если они не желают вносить свой вклад в closed-source сервисы).

Ваша **компания** может выставлять особые лицензионные требования для своих проектов. Например, можно ограничить использование вашего проекта в продуктах с закрытым исходным кодом; или – вам потребуется [копилефт-лицензия](#) и дополнительное соглашение с автором, чтобы только ваша компания (и никто другой) могла использовать ваш проект в программном обеспечении с закрытым исходным кодом; или же у вашей компании могут быть определенные потребности (связанные с какими-либо стандартами, социальной ответственностью или прозрачностью), которые могут потребовать определенной стратегии лицензирования. В любом случае, вам стоит пообщаться с юридическим отделом вашей компании.

Закрепим: когда вы создаёте новый проект на GitHub, вам предоставляется возможность выбрать лицензию. Включение одной из упомянутых выше лицензий сделает ваш проект opensource-проектом, а если вы хотите попробовать другие варианты, то просто посетите веб-сайт choosealicense.com, чтобы найти подходящую лицензию для вашего проекта (даже если проект [не связан с разработкой программного обеспечения](#)).

Что если я хочу изменить лицензию на мой проект?

В большинстве случаев, проекту никогда не понадобится замена лицензии. Но порой, обстоятельства могут измениться.

Например, по мере роста вашего проекта, появляются новые зависимости, либо же ваша компания решает сменить стратегию, а для этого может потребоваться другая лицензия. Более того – если вы не добавили лицензию на этапе запуска проекта, то добавление лицензии на данном этапе практически равнозначно её изменению. При добавлении или изменении лицензии необходимо учитывать три основных момента:

Это трудно. Определить совместимость и соответствие желаемой лицензии, а также разобраться с авторским правом, бывает крайне сложно и приводит к жуткой путанице. Простой переход на новую, но совместимую со всеми вкладами лицензию значительно отличается от процесса повторного лицензирования всех существующих вкладов. Если у вас появляется необходимость сменить лицензию, лучшим вариантом будет привлечь свою юридическую команду, потому что, даже если у вас есть разрешение от правообладателей вашего проекта на изменение лицензии или вы можете его легко получить, необходимо учитывать влияние этого изменения на других пользователей и участников вашего проекта. Воспринимайте изменение лицензии как «событие управления» для вашего проекта, организуйте связь и проконсультируйтесь со всеми заинтересованными сторонами и, скорее всего, всё пройдет гладко. Но на самом деле, лучше выбрать и использовать соответствующую лицензию на самых ранних этапах проекта!

Существующая лицензия вашего проекта. Если существующая лицензия вашего проекта совместима с той, на которую вы хотите перейти, то можно просто начать использовать новую лицензию. Причина проста: если лицензия А совместима с лицензией В, то при соблюдении условий лицензии В, условия лицензии А будут соблюдаться автоматически (в обратную сторону это правило действует не всегда). Поэтому, если вы в настоящее время используете разрешительную лицензию типа MIT, вы всегда можете перейти на более комплексную лицензию, но с тем условием, что вы сохраните копию лицензии MIT и любые связанные с ней уведомления об авторских правах. Но если на данный момент вы используете копилефт-лицензию и не являетесь единственным владельцем авторских прав, вы не можете просто так

взять и изменить лицензию своего проекта на MIT. По сути, с разрешительной лицензией правообладатели проекта дают разрешение на изменение лицензии.

Правообладатели проекта. Если вы являетесь единственным спонсором вашего проекта, то вы или ваша компания – единственный правообладатель проекта. Вы можете добавлять или изменять любую лицензию, которая вам по вкусу. В противном случае (если есть другие правообладатели), изменения лицензии необходимо согласовывать с ними. Причём учтите, что это могут быть как люди, у которых есть коммиты в вашем проекте, так и работодатели этих людей. Даже в том случае, когда люди совершают минимальные вклады – нет никаких правил, согласно которым маленькие вклады не подлежат лицензированию авторским правом. Что же делать? Тут все зависит от ситуации. Для относительно небольшого и молодого проекта, возможно, будет проще согласовать смену лицензии со всеми участниками проекта просто оформив соответствующий запрос на принятие изменений или открыв проблему. Для больших и долгоживущих проектов вам, возможно, придётся искать многих из участников или даже их наследников. Mozilla, например, потребовались годы (2001-2006), чтобы повторно лицензировать Firefox, Thunderbird и соответствующее программное обеспечение.

В качестве альтернативы, вы можете заранее указать в существующей лицензии возможность её замены при появлении определенных обстоятельств, помимо тех, которые разрешены вашей существующей лицензией с открытым исходным кодом. Это немного изменит процесс замены лицензии – вам понадобится дополнительная помощь юристов, и всё равно придётся общаться с заинтересованными сторонами вашего проекта.

Нужны ли моему проекту дополнительные соглашения?

Вполне возможно, что нет. В подавляющем большинстве проектов открытая лицензия является как входящей (inbound – от участников), так и исходящей (outbound – для других участников и пользователей). Если ваш проект размещён на GitHub, то такие условия установлены [по умолчанию](#).

Дополнительное соглашение с участниками проектов, часто называемое CLA (Contributor License Agreement), может добавить работы сопровождаителям проекта. Сколько работы будет добавлено, зависит от самого проекта и его реализации. Самый простой вариант – это когда согласно этому соглашению (извиняюсь за тавтологию) от участников требуется моментальное подтверждение того, что у них есть права, необходимые для внесения вклада в рамках лицензии. Более сложный вариант может потребовать юридической проверки и утверждения со стороны работодателей.

Кроме того, добавление «бюрократии», которая, по мнению некоторых, является совершенно ненужной, трудной для понимания или несправедливой (в том случае если бенефициар соглашения получает больше прав, чем участники или общественность), может вполне выглядеть как поведение, недружественное к сообществу проекта.



Мы ликвидировали CLA для Node.js. Это значительно снизило барьер для входа, и, соответственно, увеличило базу участников нашего проекта.

@bcantrill

Есть несколько ситуаций, когда нужно рассмотреть возможность заключения такого соглашения:

- Вашим юристам нужно, чтобы все участники принимали (подписывали, онлайн или офлайн) условия участия в проекте: может быть, потому что они считают, что самой лицензии с открытым исходным кодом недостаточно. Если это единственная причина – то достаточно подтверждения о ознакомлении с лицензией. Лицензионное соглашение [jQuery Individual Contributor](#) – отличный пример облегчённого дополнительного соглашения с участниками. Также, для некоторых проектов в качестве альтернативы может выступать [Developer Certificate of Origin](#).
- В вашем проекте используется лицензия с открытым исходным кодом, не включающая в себя явную выдачу патента (например, MIT), и вам необходим патентный грант от всех участников, некоторые из которых могут работать в компаниях с большими патентными портфелями. Такие портфели могут быть использованы в отношении вас или других участников проекта и пользователей. [Лицензионное соглашение Apache Individual Contributor](#) – наиболее часто используемое дополнительное соглашение с участниками, имеющее патентную лицензию.
- Ваш проект использует копилефт-лицензию, но вам необходимо распространять проприетарную версию проекта. Для этого вам потребуется, чтобы каждый участник предоставил вам авторские права или разрешительную непубличную лицензию. [Соглашение между участниками MongoDB](#) является примером такого типа соглашения.
- Вы думаете, что вашему проекту может потребоваться изменить лицензии в течение срока его службы и хотите, чтобы участники заранее согласились с такими изменениями.

Если вам необходимо использовать дополнительное соглашение с участниками для вашего проекта, рассмотрите возможность использовать инструменты интеграции (к примеру, [CLA assistant](#)), чтобы свести к минимуму моменты, отвлекающие участников.

Что должен знать юридический отдел моей компании?

Если вы запускаете проект с открытым исходным кодом как сотрудник компании, первое, что должен знать ваш юридический отдел – это то, что вы запускаете подобный проект.

Какой бы проект вы не запускали, лучше оповестить их, даже если это ваш личный проект. Возможно, у вас с компанией есть некое «соглашение об интеллектуальной собственности», которое даёт ей определённый контроль над вашими проектами, особенно если проекты как-то связаны с бизнесом компании или вы используете какие-либо ресурсы компании для их разработки. В целом, обычно это не является сложным процессом; если же возникнут трудности, вы всегда можете договориться (например, объяснить, что ваш проект может хорошо послужить целям профессионального обучения и развития компании), или же избегать работы над вашим проектом, пока не смените компанию.

Если вы открываете проект для своей компании, то обязательно сообщите об этом. У вашей юридической команды, вероятно, уже есть готовые и проверенные политики для использования той лицензии с открытым исходным кодом (и, возможно, дополнительного соглашения), которая основана на бизнес-требованиях компании и соответствует лицензиям возможных зависимостей проекта. Если же таких политик нет, то и вам, и вашим юристам повезло – это отличная возможность выработать их! Вот некоторые моменты, над которыми нужно поразмыслить:

- **Материалы, принадлежащие третьим лицам.** Есть ли в вашем проекте зависимости, созданные другими пользователями, или, может быть, чужой код используется каким-то иным образом? Если такие материалы имеют открытый исходный код, вам необходимо будет соблюдать правила, оговорённые в их лицензии. Для этого, стоит начать с выбора лицензии, которая работает со сторонними лицензиями с открытым исходным кодом (об этом мы писали выше). Если ваш проект будет изменять или распространять сторонние материалы, то юридическому отделу также стоит убедиться, что вы соблюдаете условия лицензий сторонних разработчиков, к примеру – сохранение уведомлений об авторских правах. Если ваш проект использует чужой код, у которого нет открытой лицензии, вам, вероятно, придётся попросить сторонних разработчиков [добавить такую лицензию](#), а в противном случае – прекратить использовать их код в своём проекте.
- **Торговые секреты.** Подумайте, есть ли в вашем проекте что-то такое, что компания не захочет делать доступным широкой публике. Если это так, то следует сначала изъять такие материалы, а уж потом – открывать исходный код проекта.
- **Патенты.** Допустим, ваша компания подаёт заявку на патент проекта, который планируется подаваться как opensource? К сожалению, вас вероятнее всего попросят подождать (или же, возможно, компания пересмотрит мудрость такого решения). Если вы ожидаете вкладов от сотрудников компаний с большими патентными портфелями, ваши юристы могут потребовать, чтобы вы использовали лицензию с явной выдачей патентов (к примеру, Apache 2.0 или GPLv3) или же дополнительное соглашение для участников.
- **Товарные знаки.** Дважды проверьте, не совпадает ли название вашего проекта с существующими торговыми марками. Если вы используете в проекте торговые марки своей компании, убедитесь, что их использование [не вызовет никаких конфликтов](#). Вам поможет [FOSSmarks](#) – это руководство разъяснит понятие товарных знаков в контексте бесплатных и открытых проектов.
- **Конфиденциальность.** Будет ли ваш проект собирать данные пользователей? В этом случае юридический отдел может помочь вам соблюсти правила компании и внешние законы.

Если вы запускаете первый проект с открытым исходным кодом в вашей компании, выполнения вышеперечисленного более чем достаточно для запуска (и уж тем более, не стоит волноваться – большинство проектов вообще не вызывают каких-либо серьёзных сложностей при запуске).

В долгосрочной перспективе ваш юридический отдел может поработать над этими вопросами более углублённо, чтобы помочь компании получить больше выгоды от участия в opensource-проектах и оставаться защищённой:

- **Политика участия сотрудников:** рассмотрите возможность разработки корпоративной политики, которая определяет, как ваши сотрудники будут участвовать в проектах с открытым исходным кодом. Чётко проработанная политика минимизирует возможную путаницу среди ваших сотрудников и поможет им свободно участвовать в проектах с открытым исходным кодом в интересах компании, будь то в рамках их работы или в их личное время. Хорошим примером является статья [Model IP and OSCP \(Open Source Contribution Policy\)](#) от Rackspace.



Выпуск интеллектуальной собственности, связанной с патчем, формирует определённую базу знаний и репутацию сотрудника. Это показывает, что компания инвестирует в развитие своих сотрудников и создаёт ощущение расширения их прав и возможностей. Все эти преимущества также приводят к повышению морального духа и лояльности сотрудников.

@vanl

- **Что стоит запускать в работу:** (почти) всё? Если ваш юридический отдел понимает стратегию компании в отношении проектов с открытым исходным кодом и инвестирует в неё, они всегда смогут вам помочь.
- **Соответствие политикам компании:** даже если ваша компания сама не запускает проекты с открытым исходным кодом, она использует программное обеспечение с открытым исходным кодом, созданное другими людьми. Осведомлённость и обработка получаемых данных могут предотвратить лишние головные боли, задержки выпуска продукта и, что самое главное – судебные процессы.

Организации должны иметь действующую стратегию лицензирования и обеспечения соответствия обоим категориям лицензий, будь то копилефт-лицензии или же разрешительные. Основа такой стратегии – тщательный учёт условий лицензирования, применяемых к используемому вами программному обеспечению с открытым исходным кодом, включая все подкомпоненты и зависимости.

Heather Meeker

- **Патенты:** вполне возможно, что ваша компания пожелает присоединиться к Open Invention Network (большой защитному пул патентов), чтобы обеспечить безопасность при использовании участниками крупных проектов с открытым исходным кодом, или же изучить другие альтернативные патентные лицензии.
- **Управление:** это особенно важно, когда имеет смысл перенести проект под управление юридического лица за пределами компании.