



Страшно ли жить с продуктом, который базируется на Open Source

Юлия Ковшова
Ярослава Лукьянова

Meet The Team

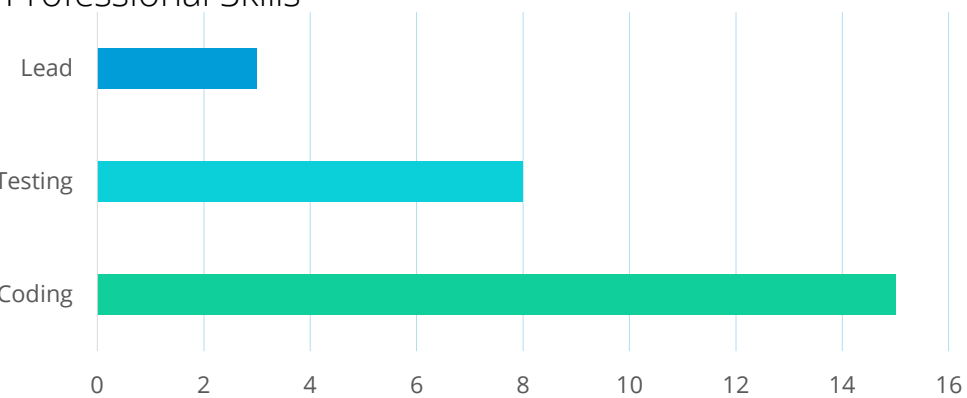


Ковшова Юлия
Управляю командой внутреннего тестирования продукта. Вечно отвечаю на вопросы зачем и для чего нам opensource.

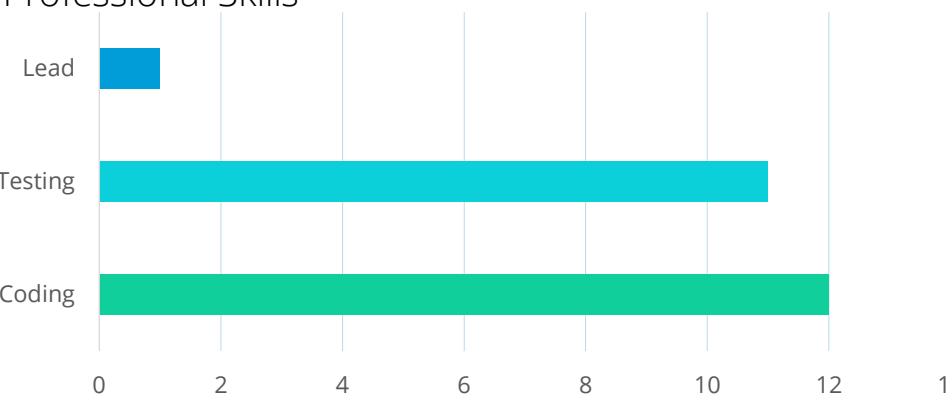


Лукоянова Ярослава
Автоматизатор тестирования со слишком большим опытом закрытого энтерпрайза за плечами.

Professional Skills



Professional Skills



Tatlin Object

1

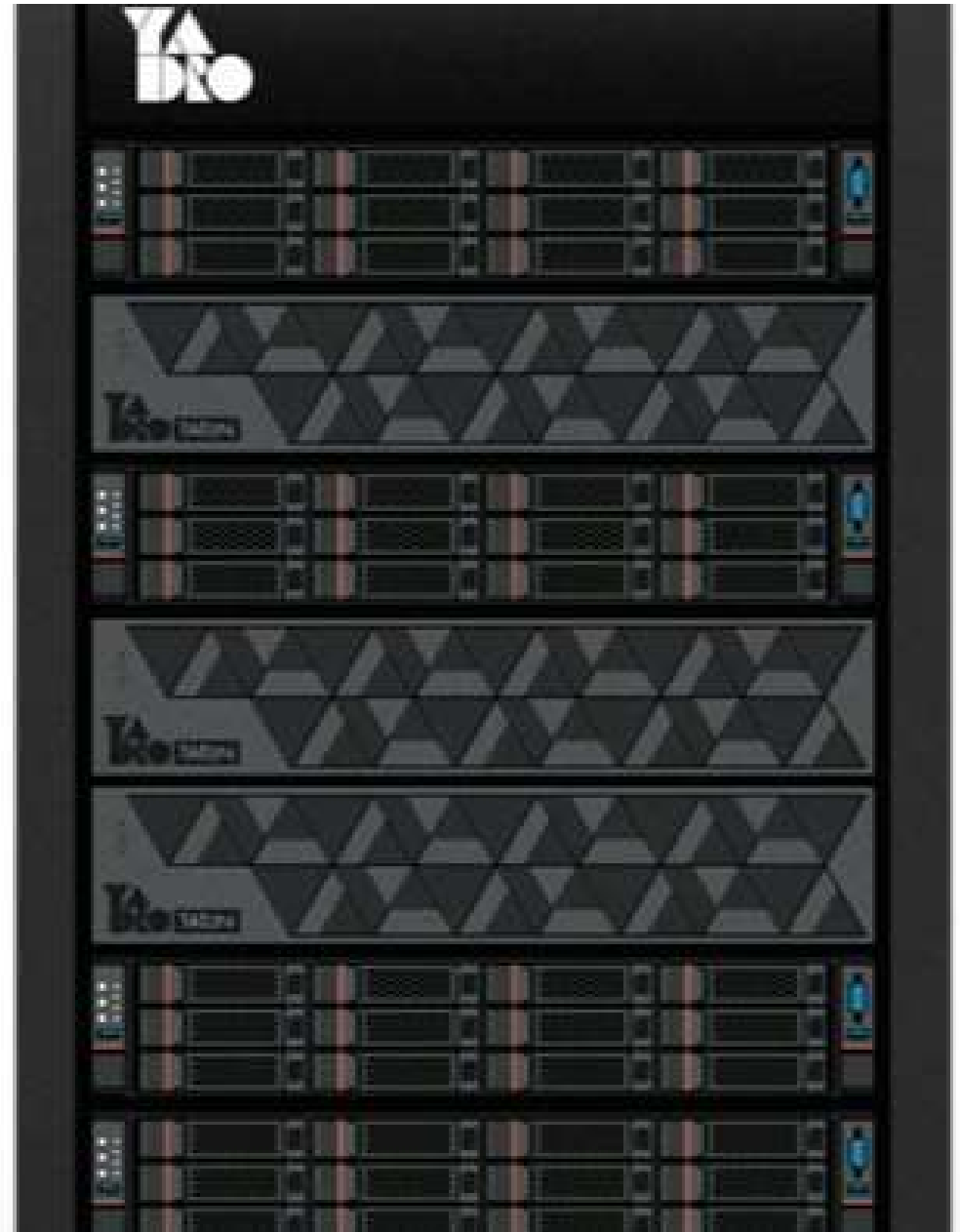
Hardware

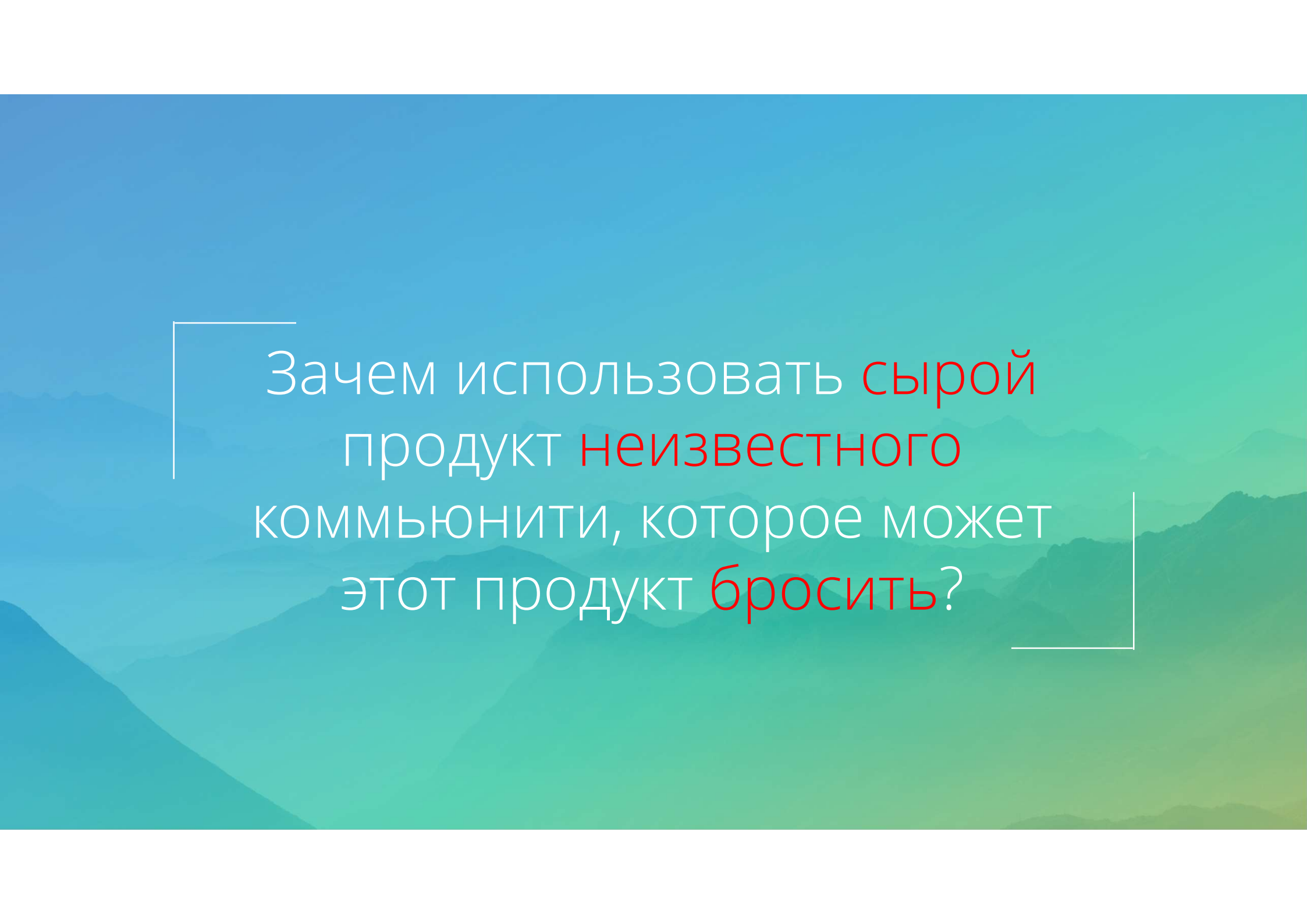
Электронные и механические части вычислительного устройства, входящие в состав системы или сети

2

Software

Программная реализация объектного хранилища данных и вспомогательные инструменты (метрики, админка, пр.)





Зачем использовать сырой
продукт неизвестного
комьюнити, которое может
этот продукт бросить?



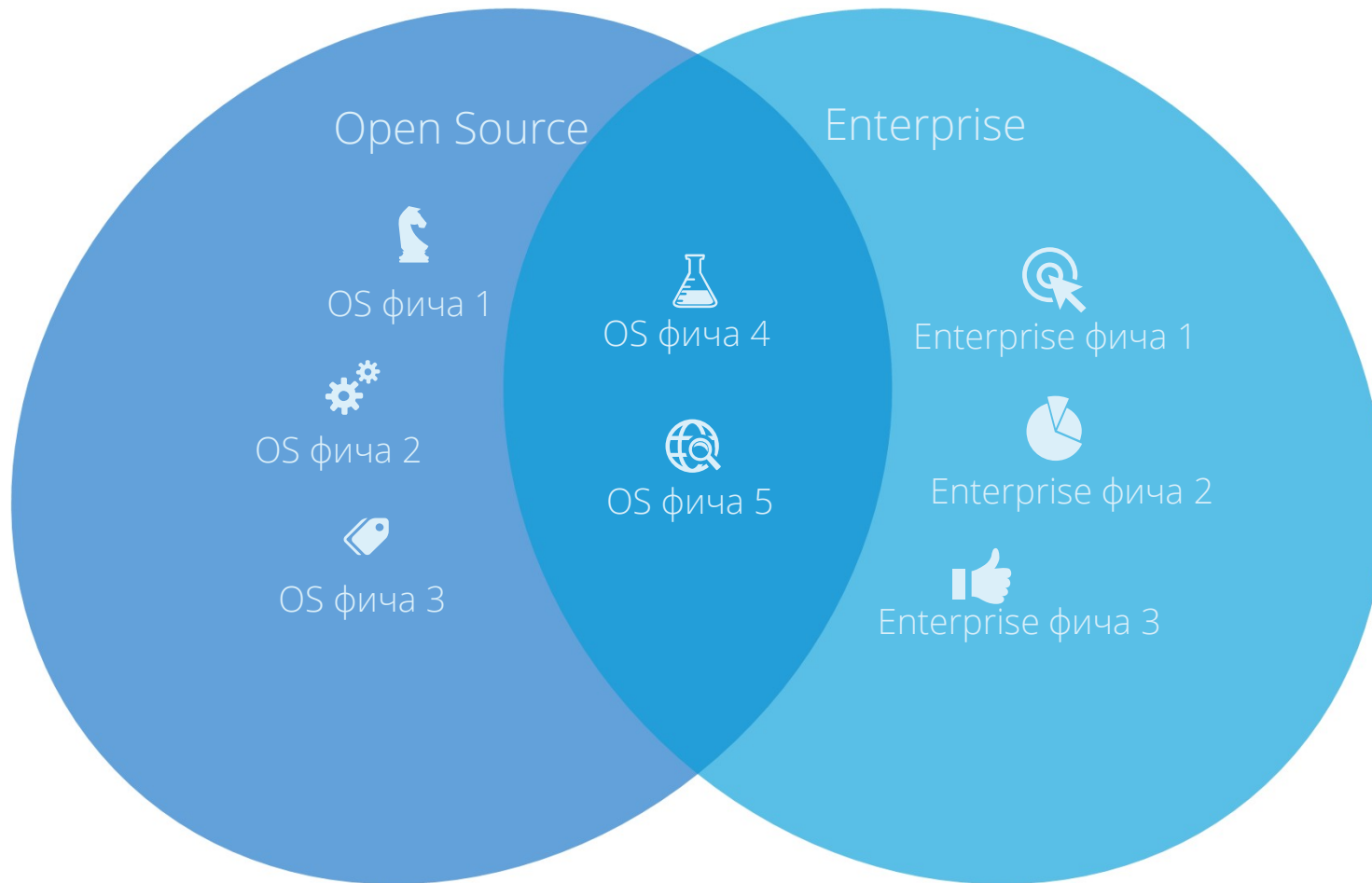
Опенсорс – просто,
вседозволенно, бесплатно?

GNU GPLv3



Кто будет покупать наш
продукт, если он равен
опенсорсу?

Классическое использование опенсорса

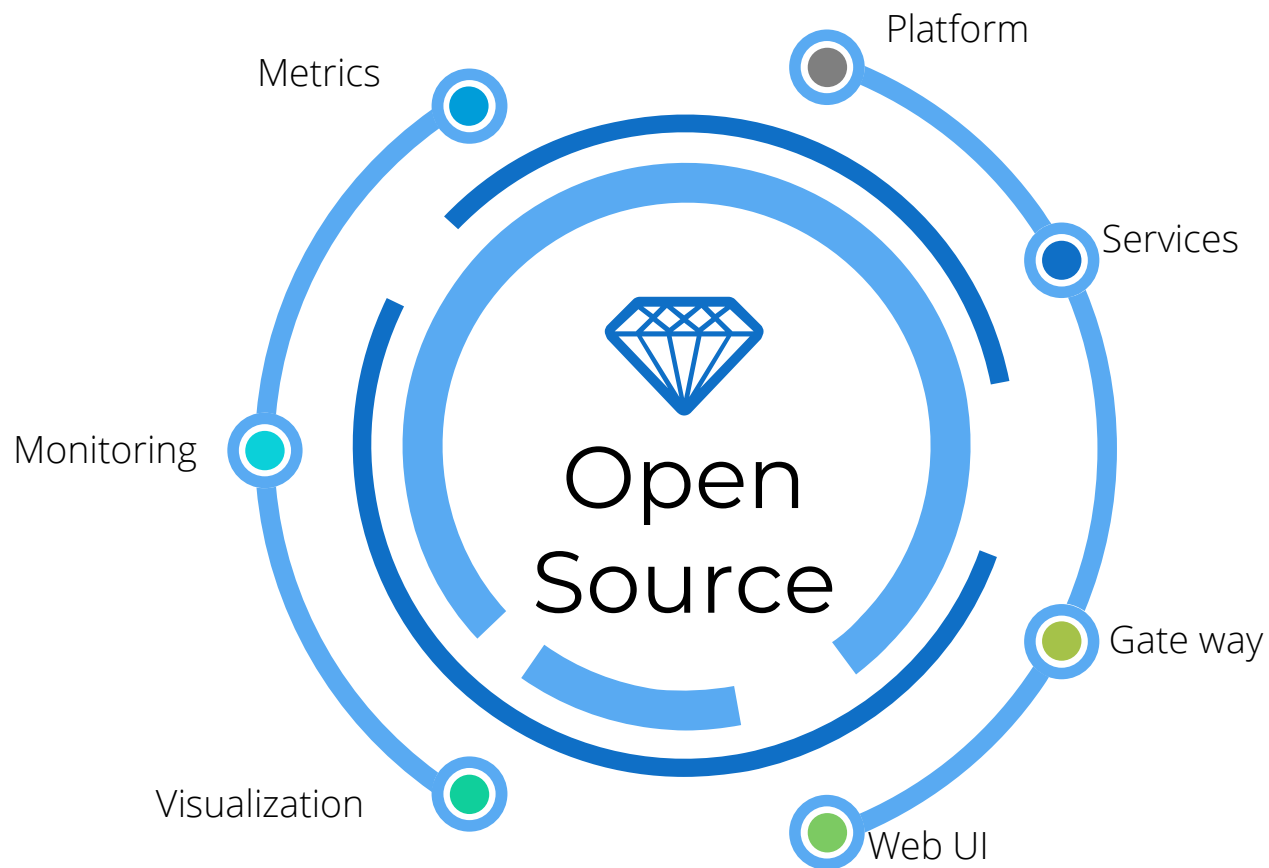


Наш опенсорс — FrostFS



Open
Source

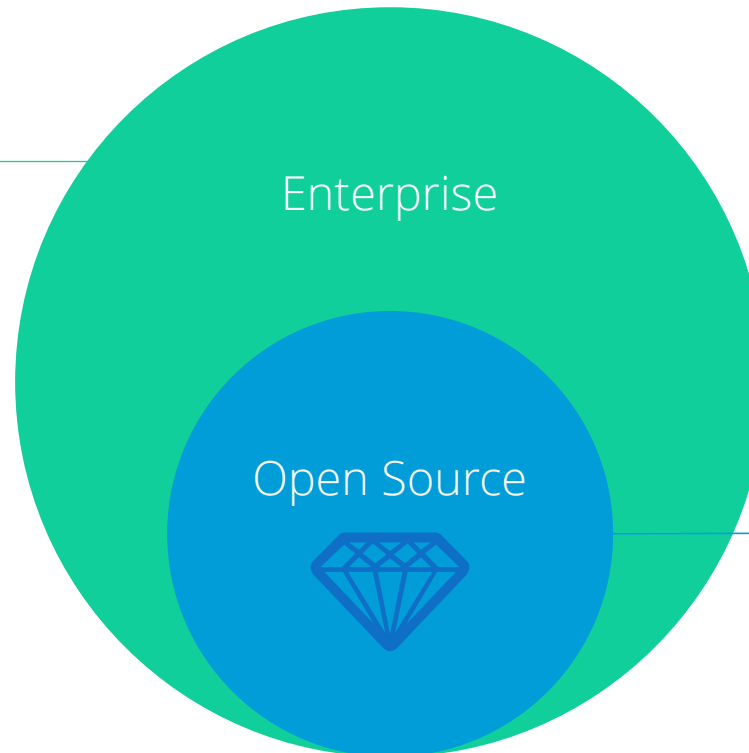
Использование опенсорса в Tatlin Object



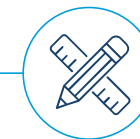
Получается, мы и тестируем
два продукта?

Тестирование

Внутреннее
тестирование
большого продукта
включает в себя
внешнее



Внешнее
тестирование
заключается только
в тестировании OS



Дизайн фреймворка для тестирования

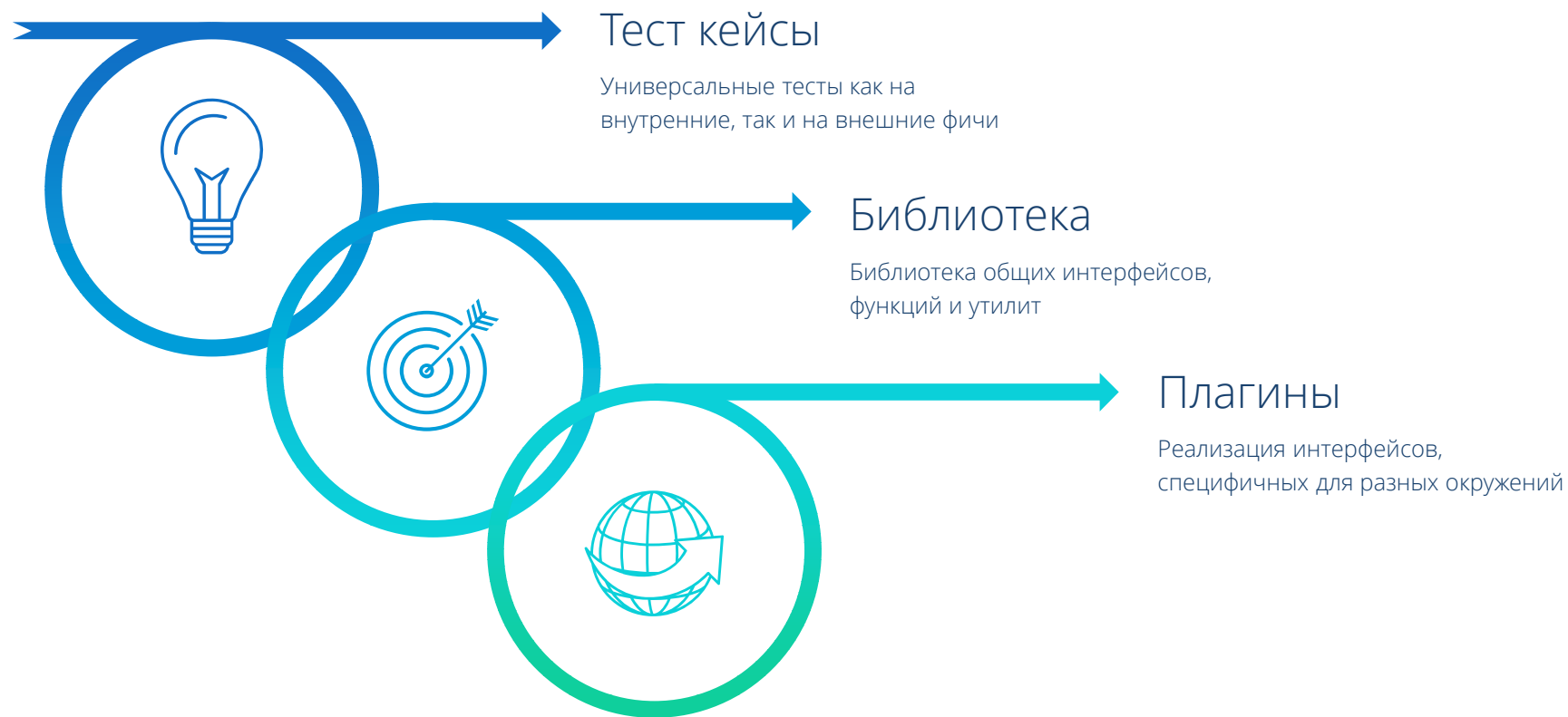
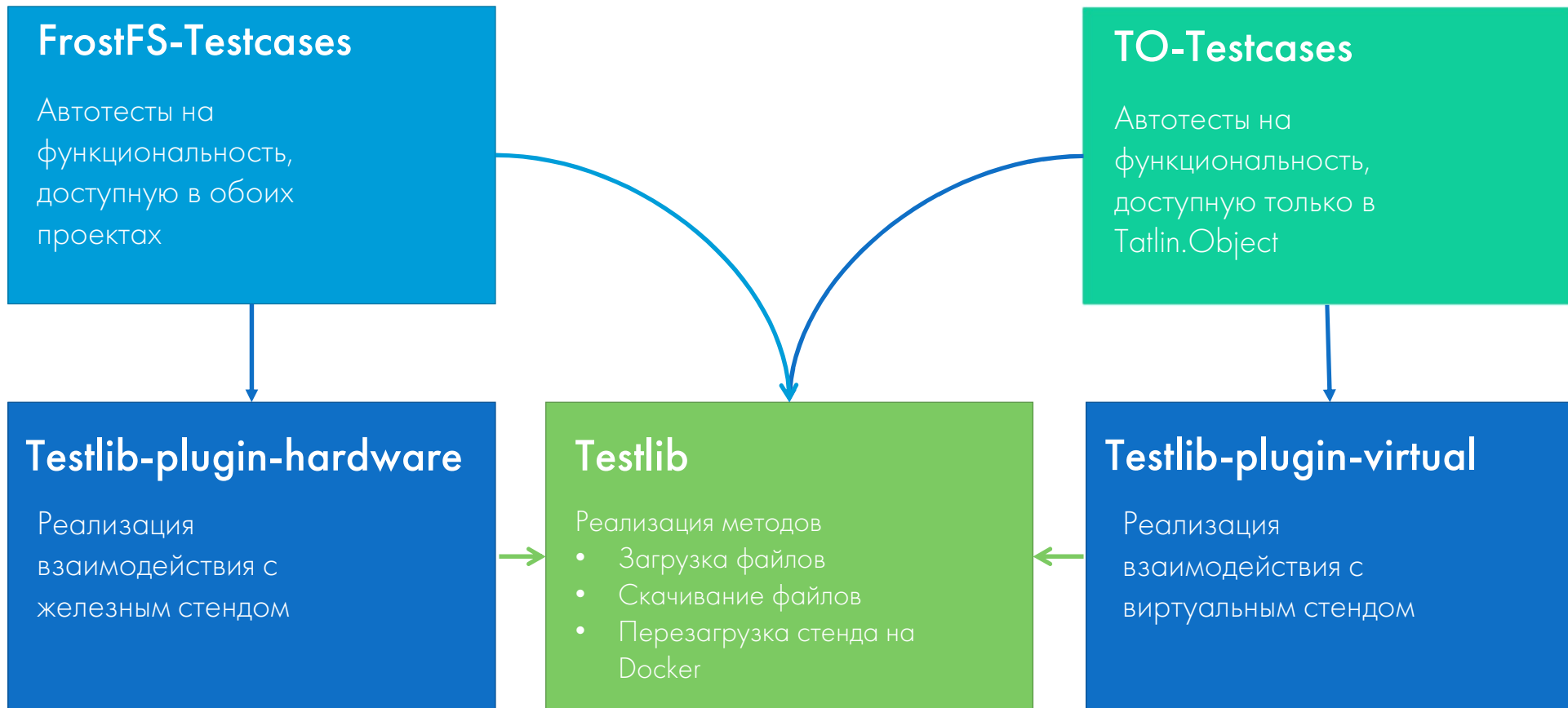


Схема устройства тестов



Внешнее тестирование

1 Внешний продукт

Автоматически собранные в отдельный продукт публичные компоненты

1

2 Внешний дженкинс

Содержим публичный CI для своевременных проверок и прозрачности

2

3 Пре-проверки

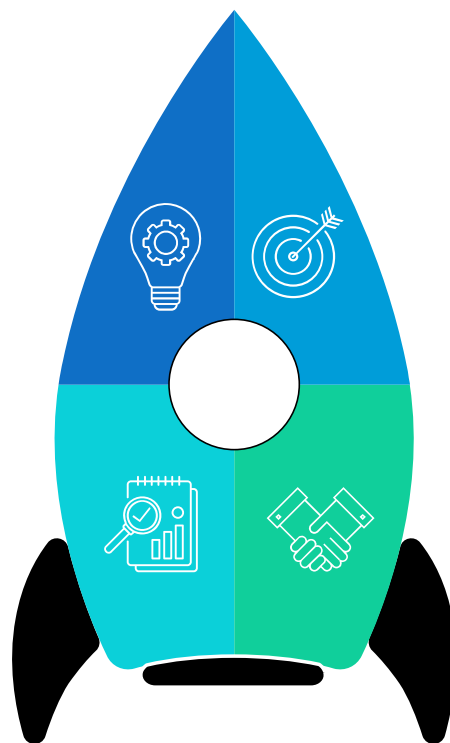
Автоматические проверки всего, что входит в опенсорс, юнит тесты

3

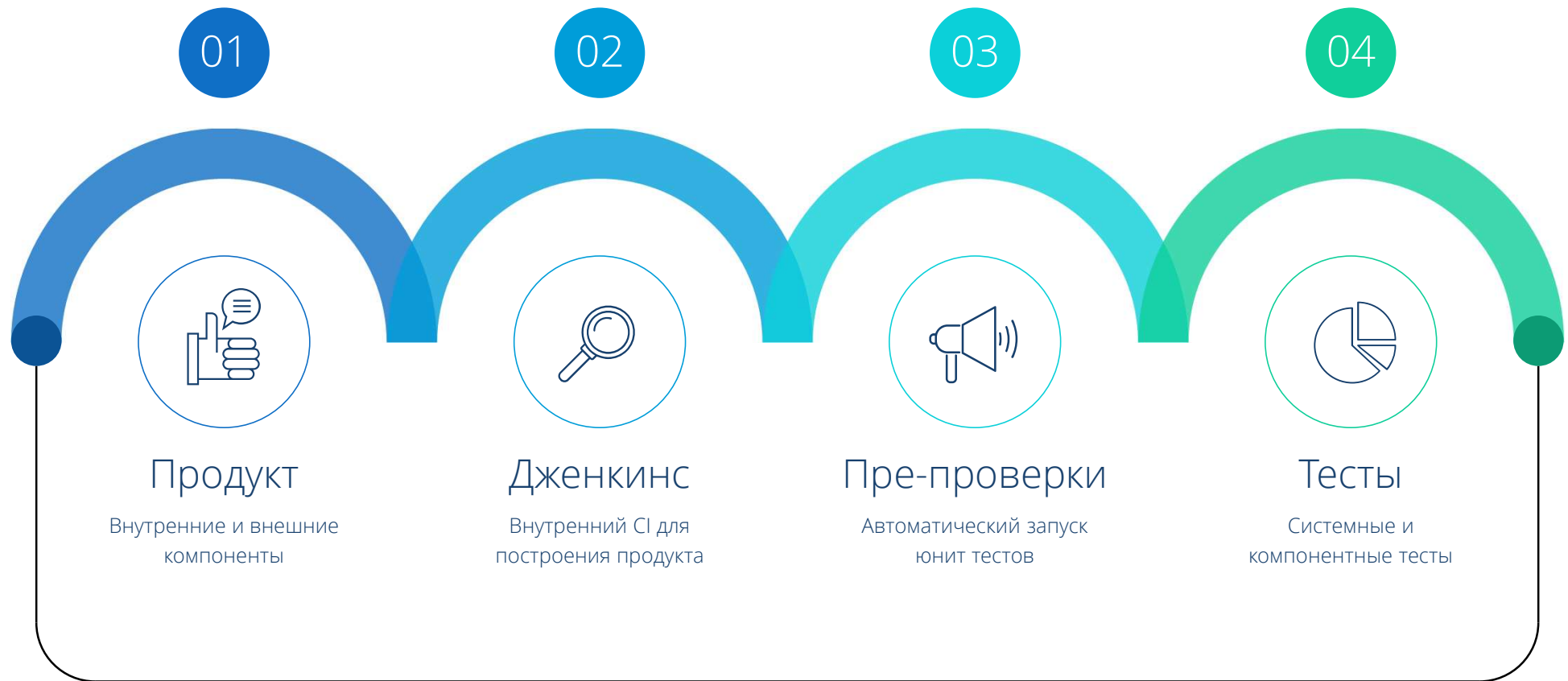
4 Тесты

Системные и компонентные тесты, отражающие качество продукта

4



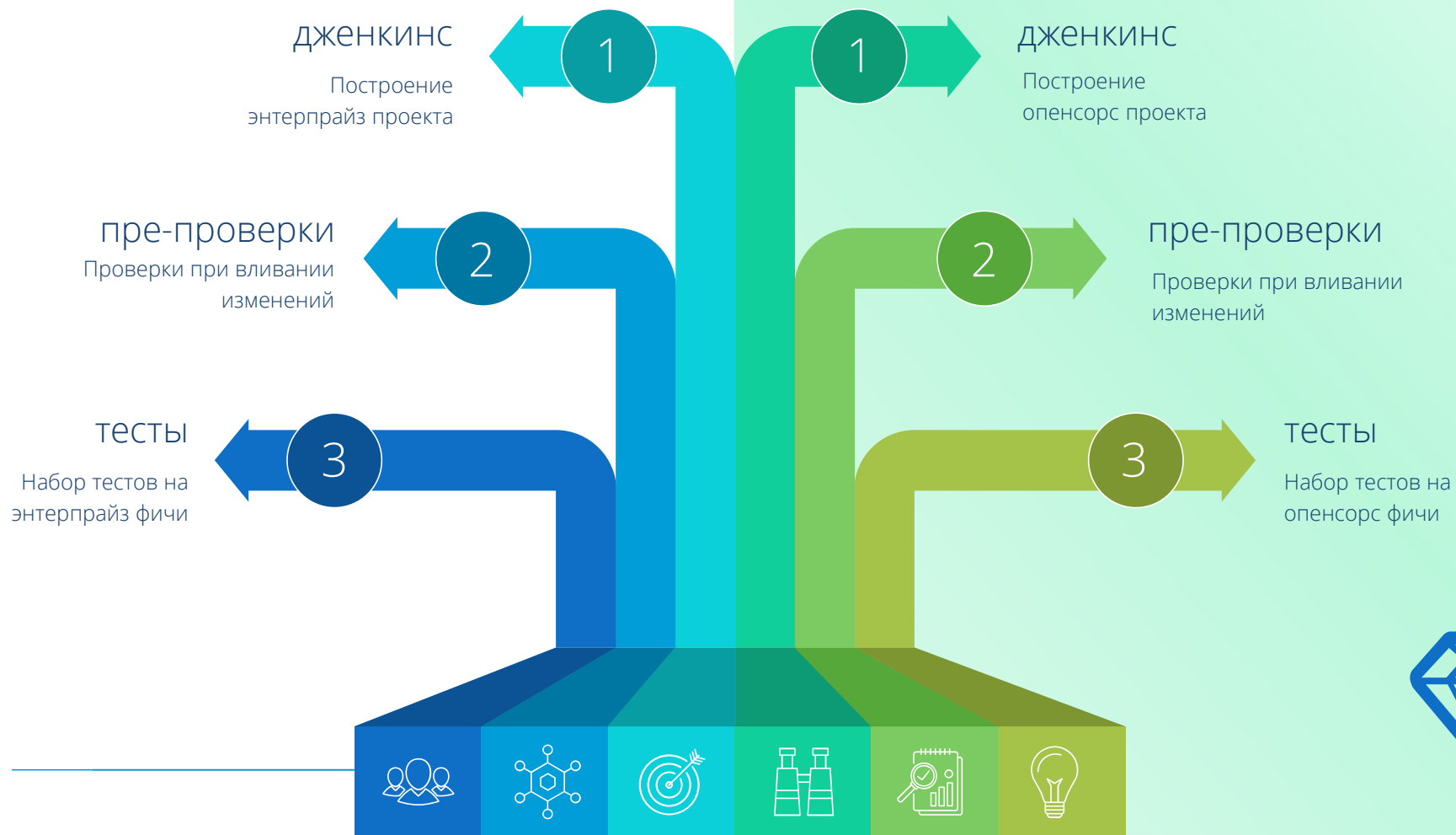
Внутреннее тестирование



Работа CI/CD

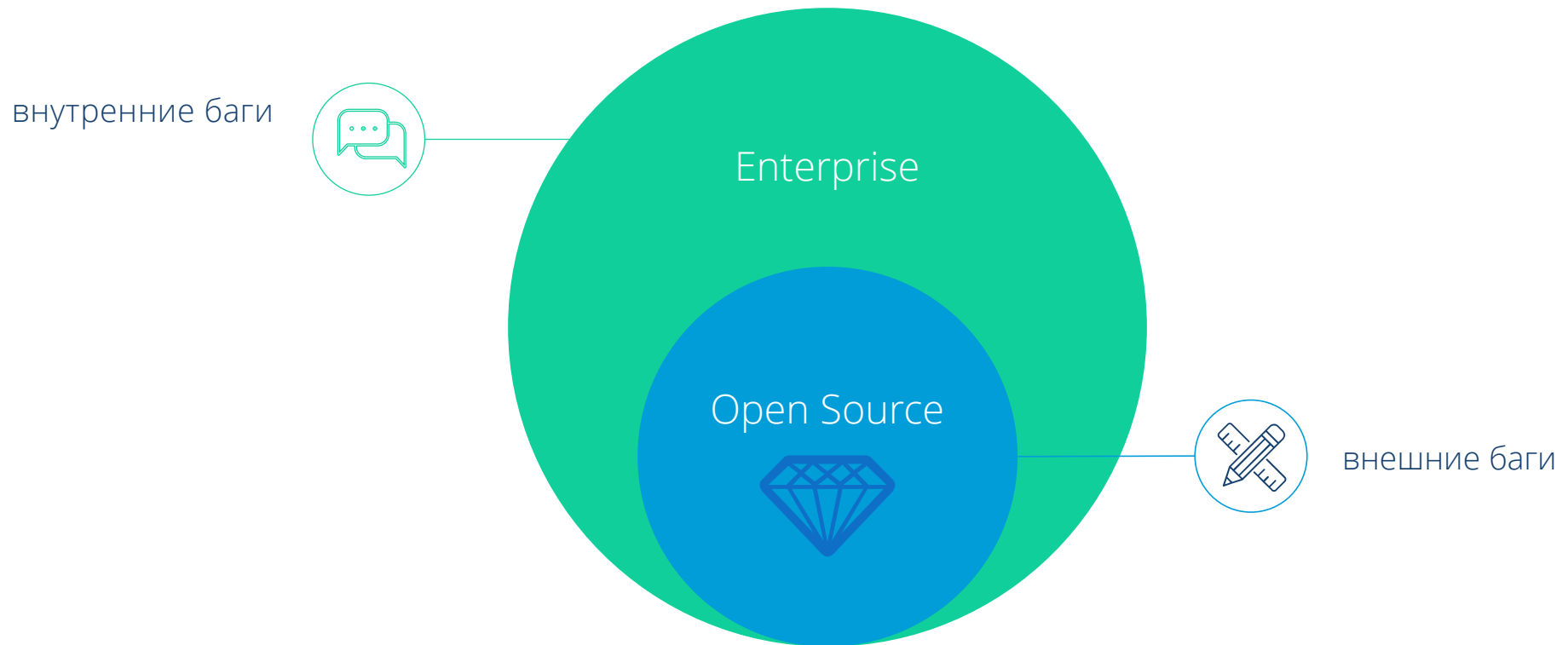
Внутренний

Внешний



Как разделять внутренние
и внешние баги?

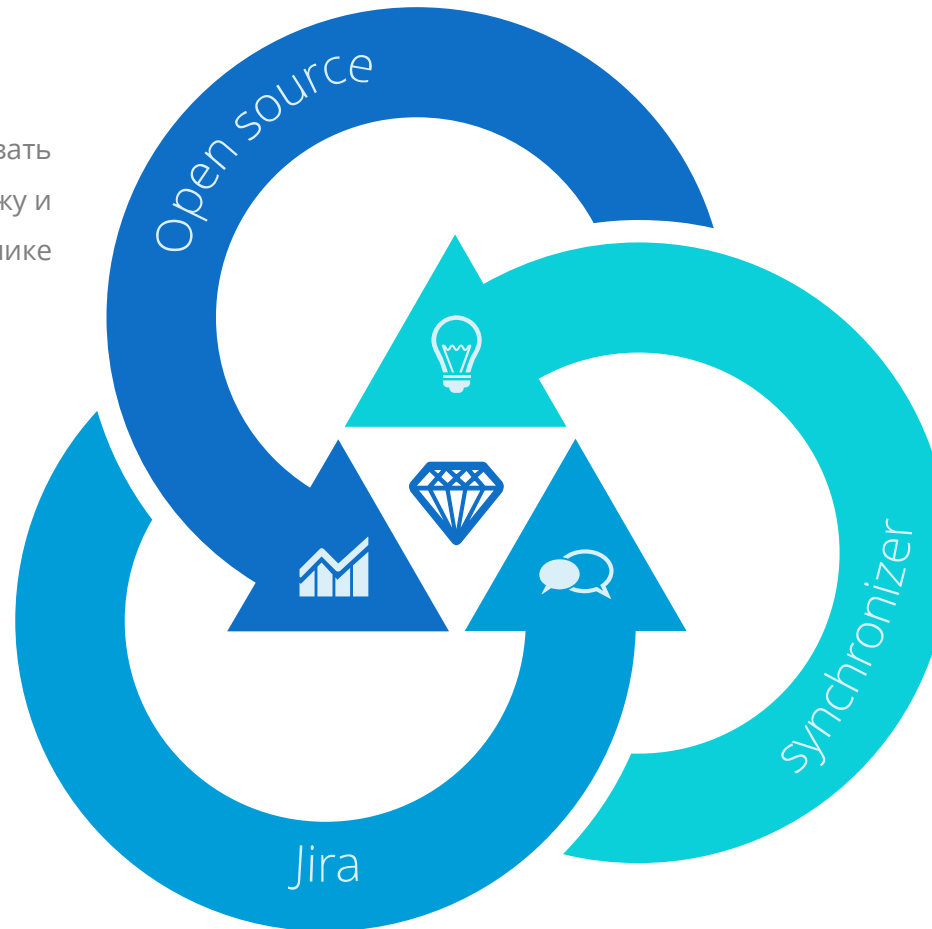
Куда заводить баги



Синхронизатор задач

Нужно показывать релевантные задачи наружу и трекать их в публице

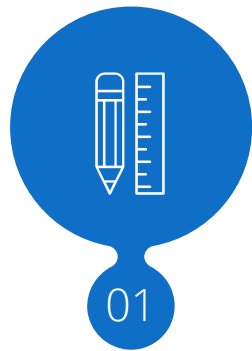
Jira позволяет вести историю задач всего продукта и видеть общий прогресс



Синхронизатор переводит все задачи из внешних во внутренние и при необходимости транслирует изменения в публик

Зеркало багов

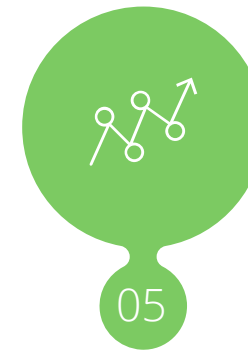
Получаем задачу на синхронизацию из Jira



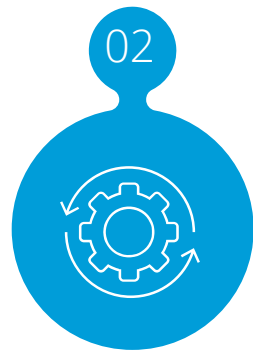
Создаём новую задачу типа «Баг» в опенсорсе



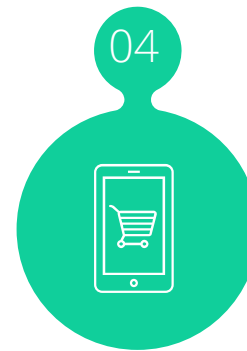
Создаем связь между внутренней и внешней задачей



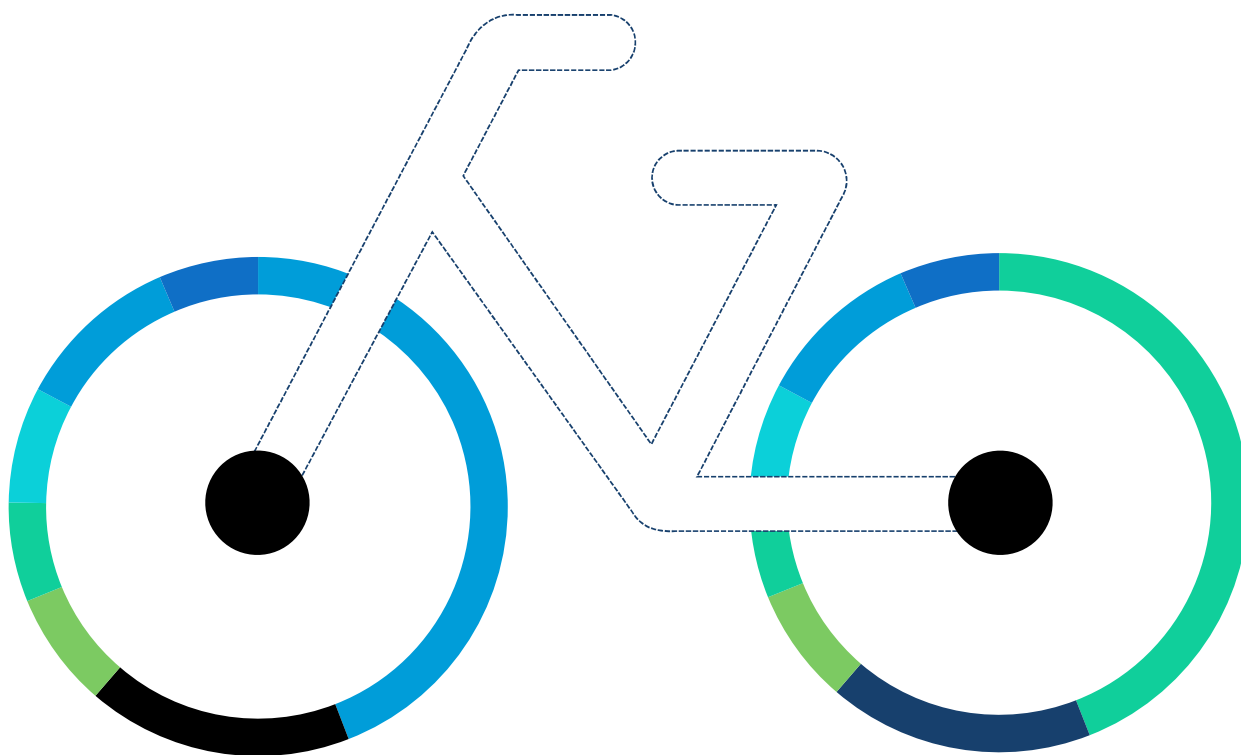
02
Определяем репозиторий опенсорса по компоненту



04
Переносим описание согласно секциям, убирая корпоративную информацию



Не только баги, но задачи




Истории

Создаются по запросам от сообщества, содержат описание новой функциональности

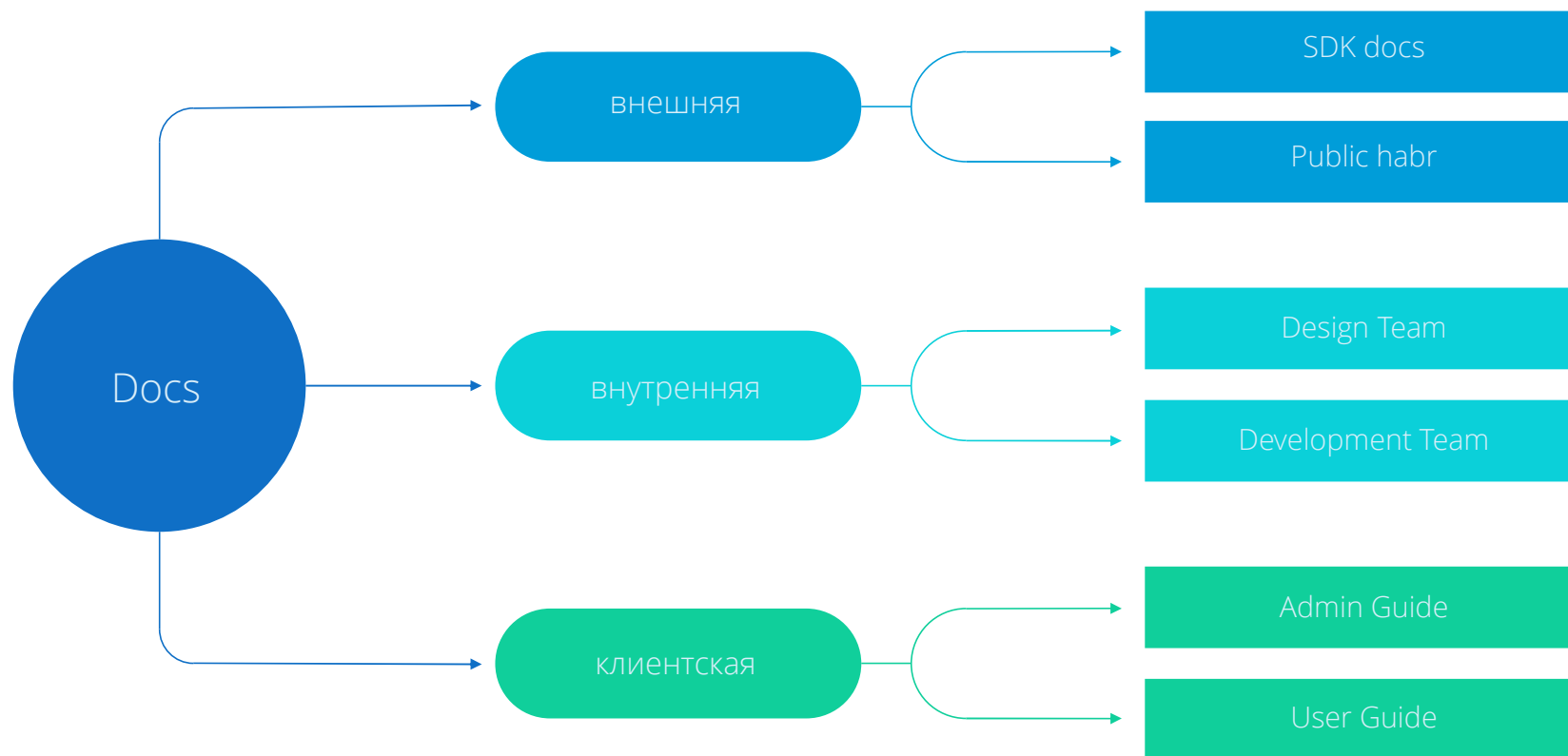
Запросы на изменения

Автоматически создаётся при изменении кода в опенсорсе без привязки к задаче.



Выходит, что и
документаций две?

Документация

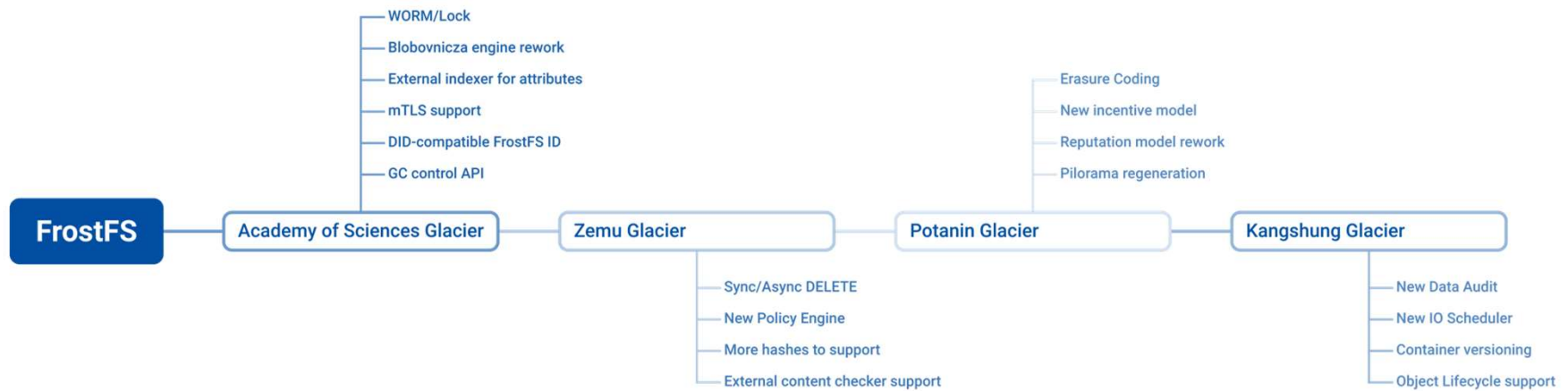


...но у нас документации недостаточно 😞



Коммьюнити может не
поддержать наш вектор
развития

Road map



Мейнтейнеры опенсорса

Минусы

Плюсы

Экстремальный корнер кейс,
который подходит не всем

Самый надёжный вариант

Свои фичи реализуются в первую очередь

Полный контроль над разработкой опенсорса

Поддерживать свой форк



Независимость от комьюнити и роудмапа опенсорса



Контроль разработки своей версии и релизного цикла



Сохранение и расширение внутренней экспертизы

Что делать с токсичностью?



быть главным заказчиком

Сложно быть токсичным с главным заказчиком своего продукта!



развивать софт скиллы

Открытость, прозрачность и умение договариваться помогают справиться с потоком токсичности



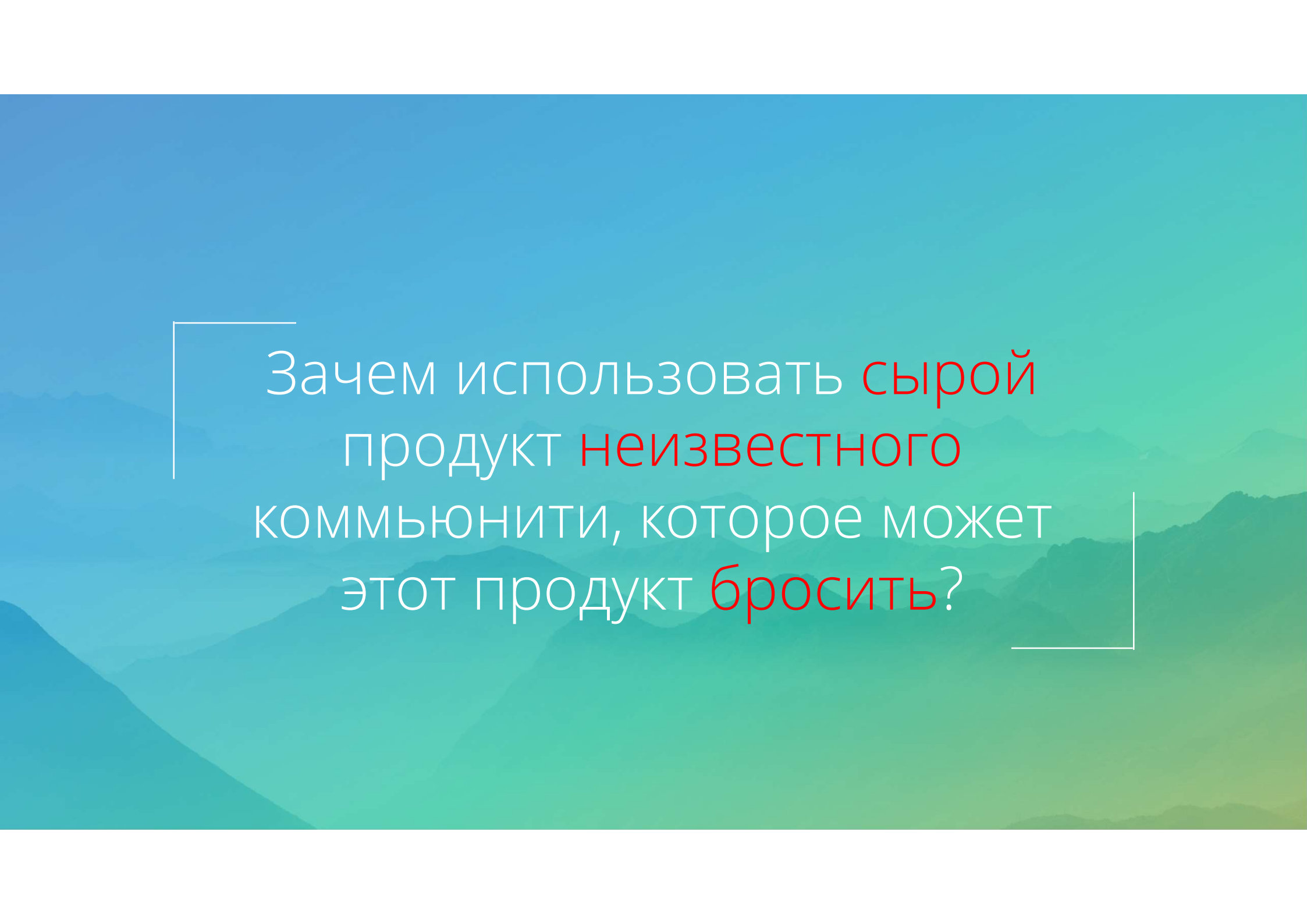
Заниматься пиаром проекта

Организация внутренних митапов и конференций увеличивает лояльность аудитории в целом

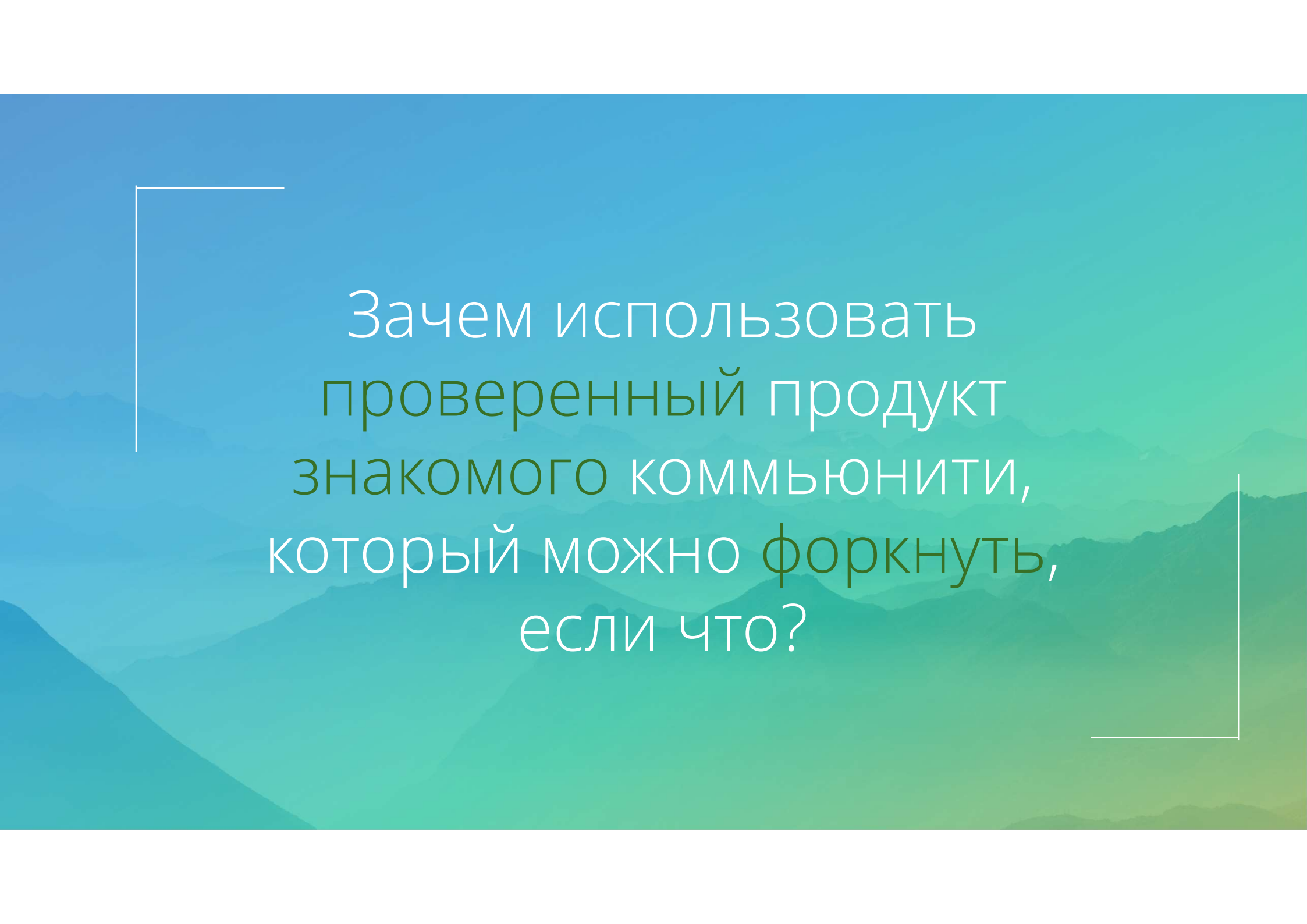




Выводы

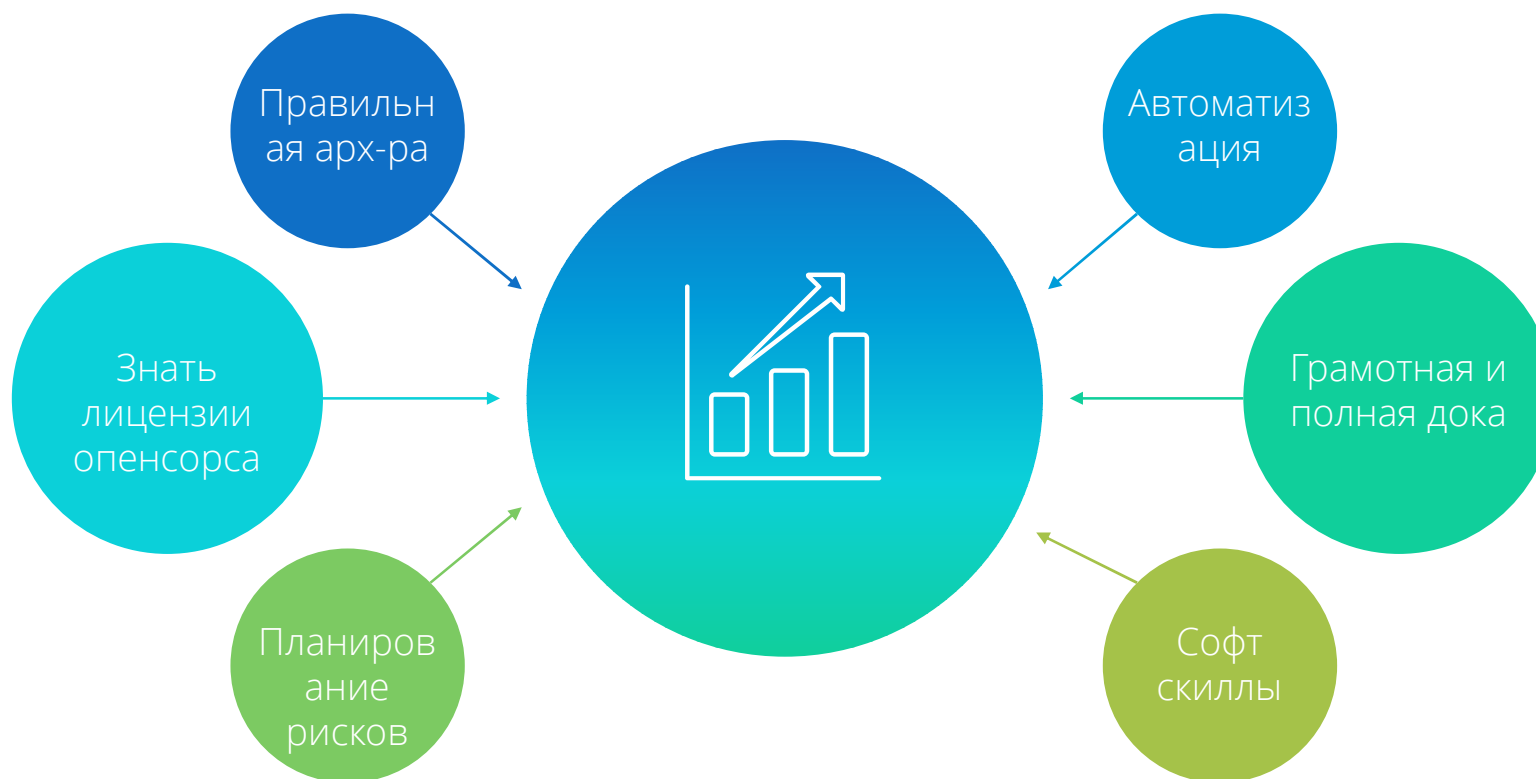


Зачем использовать сырой
продукт неизвестного
комьюнити, которое может
этот продукт бросить?



Зачем использовать
проверенный продукт
знакомого коммьюнити,
который можно форкнуть,
если что?

Ключевые факторы успеха



Thank You
For Your Watching