



ПОРА ТОРГОВАТЬ





ПОРА ТОРГОВАТЬ



ПОРА ТОРГОВАТЬ



[ MVP ]



Получите свой \*\*  
Минимальный  
жизнеспособный продукт  
\*\*, чтобы проверить свою  
концепцию.





ПОРА ТОРГОВАТЬ

[ MVP ]

\*\* периметр \*\* вашего MVP должен быть \*\* уменьшен \*\*, позволяя вам продавать ваш продукт.

Сделайте ставку на \*\* Early Adopter \*\* и получите максимум \*\* обратную связь \*\*.

Ваш MVP разворачивается и может использоваться в \*\* производстве

\*\*



ПОРА ТОРГОВАТЬ



[ СБОЙ-FAST ]



\*\* Fail \*\* fast \*\* \*\* \*\* быстро.





ПОРА ТОРГОВАТЬ

[ СБОЙ-FAST ]

\*\* Быстро испытать \*\* решение  
(несколько недель), собрать \*\*  
обратную связь \*\* от своих  
пользователей и узнать о своих  
ошибках.

Не бойтесь все менять \*\*.

Не забывайте, \*\* вы потерпите  
неудачу! \*\*



ПОРА ТОРГОВАТЬ



[ ПОЦЕЛУЙ ]



\*\* K \*\* eep \*\* I \*\* t \*\* S \*\*  
imple и \*\* S \*\* tupid.

- Почему сложнее, когда  
это может быть  
просто? \*





ПОРА ТОРГОВАТЬ

[ ПОЦЕЛУЙ ]

\*\* Избегайте чрезмерной инженерии \*\*, если достаточно «бумажной» модели или формы Google для проверки вашей концепции, не ходите дальше.

Оставайтесь просто! Технически и функционально.





ПОРА ТОРГОВАТЬ



[ ЭФФЕКТИВНОСТЬ ]

Указать меньше, \*\*  
расширить подробнее \*\*.





ПОРА ТОРГОВАТЬ

[ ЭФФЕКТИВНОСТЬ ]

Ограничьте свои спецификации до  
нужных предметов, \*\*  
сосредоточьтесь на «чем» \*\*, а не  
на «как».

Продукт должен быть  
максимально \*\*  
самодокументированным \*\*.

Документация должна быть  
версирована так же, как и код.



ПОРА ТОРГОВАТЬ



[ SAAS ]

Систематически изучать \*\*  
решения SaaS \*\*.





ПОРА ТОРГОВАТЬ

[ SAAS ]

\*\* Решения SaaS \*\* являются устойчивыми и экономически эффективными \*\*.

В некоторых случаях \*\* SaaS \*\* может \*\* ускорить \*\* реализацию \*\* MVP \*\*.

Подумайте об экономическом видении \*\* в отношении альтернатив с точки зрения \*\* общей стоимости \*\* (\*\* TCO \*\*: \*\* T \*\*otal \*\* C \*\*ost of \*\* O \*\*wnership ) и не только с точки зрения стоимости л



ПОРА ТОРГОВАТЬ



[ СЕРДЦЕ БИЗНЕСА ]

**\*\* Основной бизнес \*\*** не  
должен быть помехой для  
строительства новых услуг  
и приложений.





ПОРА ТОРГОВАТЬ

[ СЕРДЦЕ БИЗНЕСА ]

Темпы эволюции и доставки  
основного бизнеса должны быть \*\*  
совместимы с гибкостью \*\* услуг,  
которые его потребляют.

Основной бизнес должен \*\*  
предоставлять услуги \*\*.

Основной бизнес должен принять  
принцип \*\* Event-Driven \*\*, он  
сообщает о действиях  
руководства в форме событий.



ПОРА ТОРГОВАТЬ



[ НЕПРЕРЫВНОЕ  
РАЗВЕРТЫВАНИЕ ]

**\*\* Развертывание в  
производстве \*\* является  
не-событием.**





ПОРА ТОРГОВАТЬ

## [ НЕПРЕРЫВНОЕ РАЗВЕРТЫВАНИЕ ]

Воспользуйтесь \*\* непрерывным  
развертыванием \*\*, чтобы  
адаптировать \*\* \*\* производство \*\*  
к \*\* бизнес-требованиям \*\*, а не  
наоборот.

\*\* Развертывания \*\* в разных  
средах, вплоть до \*\* производства  
\*\*, должны быть \*\*  
автоматизированы \*\* и \*\* частыми  
\*\*  
.





ПОРА ТОРГОВАТЬ



[ ПЕРСПЕКТИВНАЯ БЕТА ]

\*\* Постоянный бета-метод  
\*\* позволяет привлекать  
ваших пользователей к  
процессу разработки.





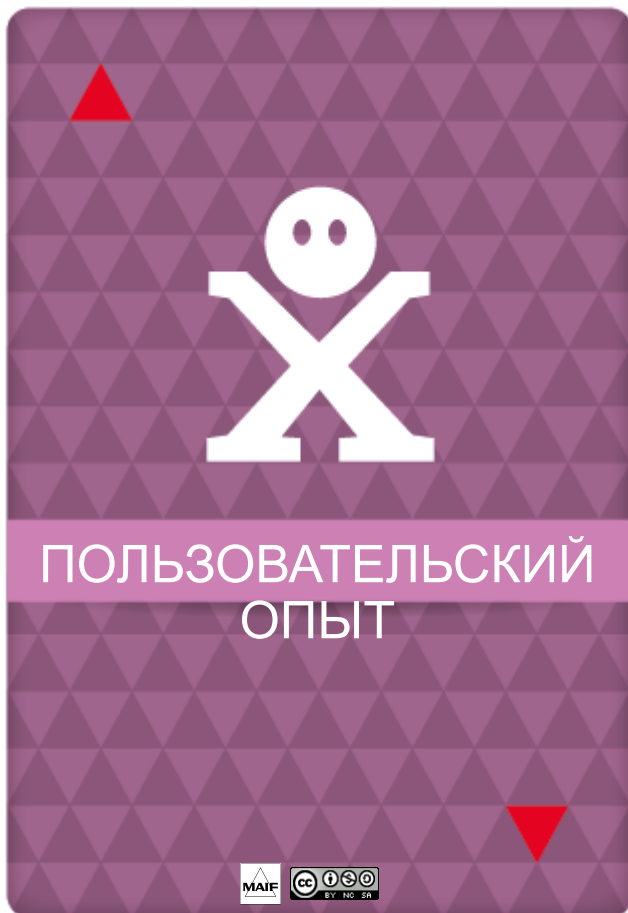
ПОРА ТОРГОВАТЬ

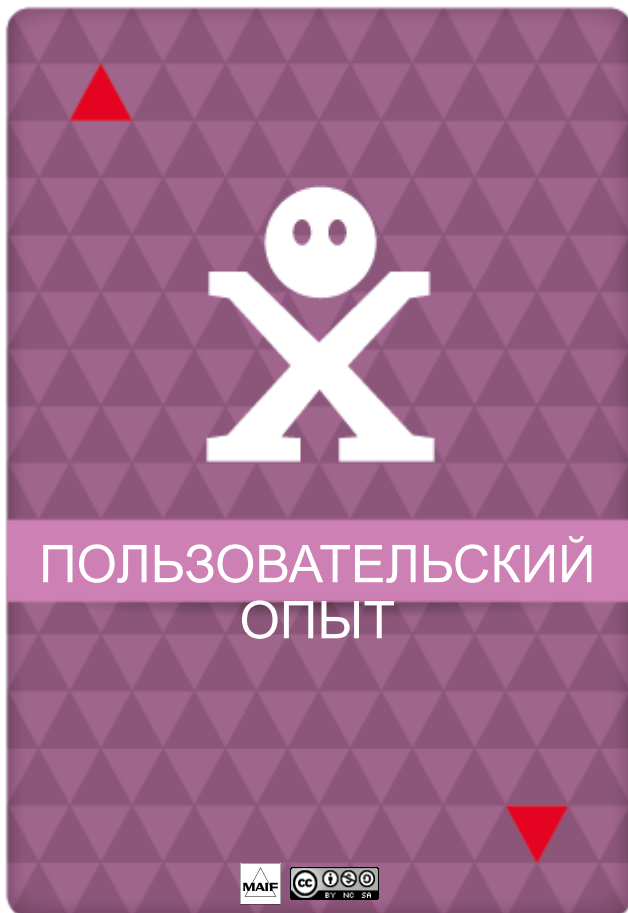
## [ ПЕРСПЕКТИВНАЯ БЕТА ]

Не стесняйтесь использовать принцип вечной беты, в котором \*\* пользователи участвуют в разработке \*\*.

Термин «вечная бета» относится к приложению, разработанному точно в срок, \*\* постоянно развивается \*\*, а не к неполному продукту.









ПОЛЬЗОВАТЕЛЬСКИЙ  
ОПЫТ

[ ВОСПРИЯТИЕ ]



\*\* Опыт, воспринимаемый  
пользователем, является  
основополагающим.

\*\* эргономика \*\* не  
подлежит обсуждению.





ПОЛЬЗОВАТЕЛЬСКИЙ  
ОПЫТ

## [ ВОСПРИЯТИЕ ]

Не пренебрегайте работой \*\*  
дизайнеров UX \*\*, это имеет  
фундаментальное значение при  
разработке приложения.

Интеграция \*\* отзывов \*\* ваших  
пользователей, это важно.





ПОЛЬЗОВАТЕЛЬСКИЙ  
ОПЫТ

[ ПРЕДСТАВЛЕНИЕ ]

Используйте \*\* мощные  
интерфейсы \*\* для  
внутреннего и внешнего  
использования.





ПОЛЬЗОВАТЕЛЬСКИЙ  
ОПЫТ

## [ ПРЕДСТАВЛЕНИЕ ]

Интерфейсы ориентированы на \*\*  
эффективность \*\*.

\*\* производительность \*\*  
интерфейса экономит время \*\*,  
увеличивает удовлетворенность  
пользователей \*\* и,  
следовательно, \*\* сохраняет их  
разочарование \*\*.







ПОЛЬЗОВАТЕЛЬСКИЙ  
ОПЫТ

[ ПЕРВАЯ МОБИЛЬНАЯ ]

Примите стратегию \*\*  
Mobile First \*\*.





ПОЛЬЗОВАТЕЛЬСКИЙ  
ОПЫТ

## [ ПЕРВАЯ МОБИЛЬНАЯ ]

Мобильные устройства являются  
\*\* самой важной \*\* частью \*\* рынка  
\*\* .

Мышление, мобильное, думает о  
\*\* существенном \*\*.

\*\* Ревизионный дизайн \*\* является  
нормой, он является источником  
сбережений (\*\* MVP \*\*).





ПОЛЬЗОВАТЕЛЬСКИЙ  
ОПЫТ

[ OMNI-КАНАЛ ]

Адаптация к  
использованию, \*\* omni-  
канал \*\* является нормой.





ПОЛЬЗОВАТЕЛЬСКИЙ  
ОПЫТ

[ OMNI-КАНАЛ ]

Многоканальный подход  
предоставляет пользователю \*\*  
объединенный опыт \*\* (пример:  
Netflix).

Различные \*\* каналы \*\* \*\*  
синхронизированы \*\* и \*\*  
когерентные \*\* (в отличие от  
пакетных процессов).

Все участники (клиенты,  
консультанты) получают доступ к  
одной и той же информации.





ПОЛЬЗОВАТЕЛЬСКИЙ  
ОПЫТ

[ SELF-DATA ]

\*\*\* пользователи \*\*  
являются \*\* владельцами  
\*\* своих данных и их курса.





ПОЛЬЗОВАТЕЛЬСКИЙ  
ОПЫТ

[ SELF-DATA ]

Оставьте \*\* индивидуумов \*\*, в  
любое время \*\*, \*\* \*\* на \*\* своих \*\*  
личных \*\* данных.

Установите \*\* уверенность \*\*,  
позволяя отслеживать и  
контролировать пользователей в  
режиме реального времени.

\*\* подсистемы \*\* должны отвечать  
тем же требованиям.





ПОЛЬЗОВАТЕЛЬСКИЙ  
ОПЫТ

[ CRM / SFA ]



Клиентские отношения  
должны быть  
унифицированы и  
контекстуализированы с  
помощью гибкого,  
унифицированного и  
управляемого событиями

\*\* CRM / SFA. \*\*





ПОЛЬЗОВАТЕЛЬСКИЙ  
ОПЫТ

[ CRM / SFA ]

Выбирайте \*\* CRM , который управляет как отношениями с клиентами, так и лидерами продаж ( SFA \*\*: \*\* S \*\*: ales \*\* F \*\*: orce \*\* A \*\*: utomation ).

\*\* CRM \*\* должен быть \*\* открыт \*\* для новых возможностей.

\*\* CRM \*\* производит \*\* события \*\*, соответствующие действиям руководства, вписывающимся в логику \*\* платформы Event-Driven

\*\*







ПОЛЬЗОВАТЕЛЬСКИЙ  
ОПЫТ

[ БОЛЬШИЕ ДАННЫЕ ]

Платформа Big Data  
позволяет централизовать  
\*\* и обрабатывать  
пользовательские данные,  
чтобы наилучшим  
образом обслуживать \*\*  
путешествие \*\*.





ПОЛЬЗОВАТЕЛЬСКИЙ  
ОПЫТ

## [ БОЛЬШИЕ ДАННЫЕ ]

Централизовать \*\* Группу Maif \*\*, \*\*  
Партнер \*\* и \*\* Данные поставщика  
\*\* в логике \*\* пути \*\*.

«Подготовка данных» и обработка  
могут \*\* консолидировать \*\*  
данные.

\*\* Команды «Большие данные»  
сотрудничают \*\* с  
функциональными группами для  
обеспечения \*\* данных \*\*  
управления.





ПОЛЬЗОВАТЕЛЬСКИЙ  
ОПЫТ

[ DESKTOP ]

Рабочая станция  
адаптирована и  
адаптируется к \*\*  
использует \*\* и \*\*  
современные каналы \*\*.





ПОЛЬЗОВАТЕЛЬСКИЙ  
ОПЫТ

[ DESKTOP ]

Примите \*\* федерацию  
идентичности \*\* для  
унифицированного опыта.

\*\* \*\* \*\* \*\* \*\* позволяет \*\* предлагать \*\*  
обзор \*\*, он не заменяет  
приложения.

Рабочая станция должна быть \*\*  
мобильной \*\*, \*\* многоканальной \*\*  
и \*\* стандартной \*\*, чтобы  
разрешить открытие в \*\*  
расширенном предприятии \*\*.





ПОЛЬЗОВАТЕЛЬСКИЙ  
ОПЫТ

[ АВТОРЫ ]

Не забывайте, что ваши \*\*  
партнеры \*\* используют  
современные приложения  
дома в UX.





ПОЛЬЗОВАТЕЛЬСКИЙ  
ОПЫТ

[ АВТОРЫ ]

Лечите \*\* всех своих  
пользователей как «клиентов» \*\*:   
пользователи Интернета,  
менеджеры, операторы,  
разработчики и т. Д.

Не стоит недооценивать \*\* усилие  
UX \*\* для внедрения приложений  
управления внутренним  
использованием.





ПОЛЬЗОВАТЕЛЬСКИЙ  
ОПЫТ

[ ВСЕ ИЗМЕРЕНИЕ ]



Все, что можно измерить,  
должно быть.

\*\* Без меры все это только  
мнение \*\*.





ПОЛЬЗОВАТЕЛЬСКИЙ  
ОПЫТ

[ ВСЕ ИЗМЕРЕНИЕ ]

Подумайте о показателях во  
время \*\* разработки \*\*  
приложения. \*\* журналы \*\* должны  
иметь \*\* бизнес, а также  
технические \*\* измерения.

Не пренебрегайте  
характеристиками \*\* \*\*, они  
фундаментальны.

Функциональная группа  
обеспечивает \*\* операцию \*\*: она  
отвечает за использование \*\*  
приложения \*\*







ПОЛЬЗОВАТЕЛЬСКИЙ  
ОПЫТ

[ Тестирование А / В ]

**\*\* Тестирование А / В \*\***  
экономит ваше время,  
позволяя решить \*\*  
обратную связь \*\*.





ПОЛЬЗОВАТЕЛЬСКИЙ  
ОПЫТ

## [ Тестирование А / В ]

Вместо того, чтобы произвольно решать два решения, не стесняйтесь настраивать \*\* А / В тестирование \*\*.

Этот шаблон состоит из представления \*\* двух разных версий \*\* одного и того же приложения и выбора одного из них на основе \*\* объективных мер \*\* активности пользователя.





ПОЛЬЗОВАТЕЛЬСКИЙ  
ОПЫТ

[ ИЗНОС ]



**\*\* Учитывайте деградацию  
\*\*, а не прерывание  
обслуживания в случае  
сбоя.**





ПОЛЬЗОВАТЕЛЬСКИЙ  
ОПЫТ

## [ ИЗНОС ]

В \*\* сбое \*\* одной из подсистем \*\* ненормальная \*\* версия службы должна \*\* считаться \*\* в первую очередь, а не прерыванием.

С \*\* Автоматическими выключателями \*\*, \*\* изолируйте пробой \*\*, чтобы \*\* избежать \*\* его \*\* удара \*\* и \*\* распространения \*\* по всей \*\* системе \*\*.





ЧЕЛОВЕЧЕСКИЙ





ЧЕЛОВЕЧЕСКИЙ



ЧЕЛОВЕЧЕСКИЙ



[ КОММЕНТАРИЙ  
FEATURE ]



Команда организована  
вокруг \*\* продукта \*\* или \*\*  
услуги \*\*, оказанной.





ЧЕЛОВЕЧЕСКИЙ

[ КОММЕНТАРИЙ  
FEATURE ]

Команды - \*\* Feature Feature \*\*, организованные вокруг согласованного функционального набора и состоящие из всех \*\* навыков \*\*, необходимых для этого набора.

Например: Business Expert + Web Developer + Java Developer + Architect + DBA + Operational.

\*\* Ответственность \*\* является \*\* коллективной \*\*, функциональная группа обладает полномочиями, необходимыми для этой





ЧЕЛОВЕЧЕСКИЙ



[ КОМАНДА 2-ПИЦЦЫ ]

Ограничьте \*\* размер  
функциональных групп \*\*  
(от 5 до 12 человек).





ЧЕЛОВЕЧЕСКИЙ

[ КОМАНДА 2-ПИЦЦЫ ]

Ограничьте размер функциональной группы: \*\* от 5 до 12 человек \*\*.

Ниже 5 она слишком чувствительна к внешним событиям и испытывает недостаток в творчестве. Выше 12, он теряет производительность.

Термин \*\* «2-Pizza Team» \*\* указывает, что размер Feature Team не должен превышать количество людей, которым можно кормить двумя



BY NC SA

ЧЕЛОВЕЧЕСКИЙ



[ ПРОГРАММНОЕ  
ОБЕСПЕЧЕНИЕ ARTISAN  
]



Ставка на универсальных  
людей, которые \*\* умеют  
делать \*\* и которые \*\*  
любят делать \*\*.





ЧЕЛОВЕЧЕСКИЙ

## [ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ARTISAN ]

Самой важной является \*\*  
культура развития \*\*, \*\*  
масштабируемость \*\* и \*\*  
адаптивность \*\*.

Рекрутинг \*\* разработчиков  
программного обеспечения и  
разработчиков полных стеков \*\*,  
они приносят реальную  
добавленную стоимость  
благодаря их ноу-хау и их общему  
видению.



Тем не менее, мобильные

ЧЕЛОВЕЧЕСКИЙ



[ КАДРОВЫЙ ]

Будьте \*\* привлекательны  
\*\*, чтобы набрать \*\*  
лучший \*\*.





ЧЕЛОВЕЧЕСКИЙ

[ КАДРОВЫЙ ]

Предложить рабочие режимы,  
адаптированные к сотрудникам: \*\*  
мобильность \*\*, \*\* домашняя  
работа \*\*, \*CYOD\* (Choose Your  
Own Device).

Дайте время для экспериментов и  
сделайте это \*\* в рабочее время \*\*.



ЧЕЛОВЕЧЕСКИЙ



[ ЕВА ]

Организация должна быть  
\*\* двигателем сна \*\*

Накануне это часть  
работы.





ЧЕЛОВЕЧЕСКИЙ

[ EVA ]

Организация должна быть \*\*  
дневным движком \*\*, создавая  
такие системы, как \*\* непрерывное  
образование \*\* или \*\* бизнес-  
университеты \*\*.

Не стесняйтесь сочетать их с  
другими более неформальными \*\*  
способами, такими как: \*\*  
Кодирование Dojos \*\*, \*\* Кофейные  
обеды \*\*, \*\* Внешние \*\*  
Конференции.





ЧЕЛОВЕЧЕСКИЙ



[ СО-КОНСТРУКЦИЯ ]

Преодолейте барьеры  
между сделками, сделайте  
ставку на цели \*\*  
конвергенции \*\*.





ЧЕЛОВЕЧЕСКИЙ

[ СО-КОНСТРУКЦИЯ ]

Чтобы преодолеть барьеры между сделками, недостаточно объединить людей вокруг общего продукта в общем месте.

**\*\* Гибкие подходы \*\*** устраняют эти барьеры, чтобы обеспечить **\*\* сближение целей \*\***.

Эти методы являются неотъемлемой частью ключей к успеху, организация является гарантом.



ЧЕЛОВЕЧЕСКИЙ



[ DevOps ]

**\*\* Методы DevOps \*\***  
**позволяют стенам упасть**  
**между Build и Run.**





ЧЕЛОВЕЧЕСКИЙ

[ DevOps ]

Принять \*\* DevOps \*\* для  
сближения \*\* Dev \*\* и \*\* Ops \*\* в  
направлении общей цели: \*\*  
служить организации \*\*.

\*\* Торги остаются разными \*\*!  
DevOps не означает, что один и  
тот же человек выполняет задачи  
Dev и Ops. \*\* Разработчики \*\* и \*\*  
Операционные \*\* обязаны \*\*  
сотрудничать \*\*, чтобы \*\* извлечь  
выгоду \*\* из \*\* \*\* навыков \*\* и  
улучшить \*\* эмпатию \*\*.



ЧЕЛОВЕЧЕСКИЙ



[ PAIN ]



**\*\* Жесткие задания \*\***  
**выполняются \*\* Группой**  
**разработчиков \*\*.**

Затем следует  
автоматизация.





ЧЕЛОВЕЧЕСКИЙ

[ PAIN ]

В традиционной организации \*\*  
отсутствие понимания \*\* между  
командами обычно связано с  
расстоянием и \*\* отсутствием  
связи \*\*.

\*\* Члены функциональной группы  
\*\* \*\* несут ответственность \*\* и \*\*  
солидарны \*\* для всех задач.

\*\* Боли \*\* - ключевой фактор в \*\*  
Непрерывном улучшении \*\*.



ЧЕЛОВЕЧЕСКИЙ



[ CDS ]

Сервисные центры трудно  
согласовать с \*\*  
коллективным  
обязательством \*\*.





ЧЕЛОВЕЧЕСКИЙ

[ CDS ]

Функциональные группы  
построены на принципах, которые  
в значительной степени зависят от  
сотрудничества \*\* и \*\*  
коллективного участия \*\*.

Сервисные центры продвигаются  
к рационализации и консолидации  
ИТ по бизнесу, что противоречит  
этому понятию коллективной  
ответственности.





ЧЕЛОВЕЧЕСКИЙ



[ ПРОВЕРКА ]

Организация имеет \*\* роль  
валидации \*\*, не будучи  
догматической.





ЧЕЛОВЕЧЕСКИЙ

[ ПРОВЕРКА ]

Убедитесь, что организация сохраняет свою \*\* роль проверки \*\* в инструментах и целях использования. В частности, в \*\* инструментах, влияющих на наследие \*\* (пример: управление исходным кодом).

\*\* Предоставить \*\*

Функциональные группы с \*\* означает \*\* для поддержки их выбора.

Не будьте \*\* догматичны \*\* и обязательно  айте

ЧЕЛОВЕЧЕСКИЙ



[ Трансверсальности ]



Ожидается, что группы  
функций \*\* будут общаться  
\*\* и поделиться своим \*\*  
опытом \*\* и \*\* навыками \*\*.





ЧЕЛОВЕЧЕСКИЙ

[ Трансверсальности ]

Не создавайте препятствий между  
\*\* Feature Groups \*\*.

Настройте \*\* организацию \*\* и \*\*  
ловкость \*\*, необходимую для  
функциональных групп, чтобы  
общаться друг с другом и  
делиться своими навыками и  
опытом.

Организация трансверсальности в  
\*\* Spotify \*\* (Tribes, Chapters and  
Guilds) является красноречивым  
примером.





# СОВМЕСТИМОСТЬ





# СОВМЕСТИМОСТЬ



## СОВМЕСТИМОСТЬ



[ API для всех ]



**\*\* API для всех видов  
использования \*\*:  
внутренние, клиенты и  
партнеры, публичные.**





## СОВМЕСТИМОСТЬ

[ API для всех ]

Откройте свою организацию для новых пользователей и новых клиентов с \*\* Public API \*\*.

В \*\* коммерческих \*\* партнерствах \*\* , \*\* клиенты \*\* как \*\* провайдеры \*\* , API являются стандартным обменным форматом.

\*\* API \*\* также предназначены для использования \*\* для внутреннего использования \*\* организации.





СОВМЕСТИМОСТЬ



[ САМООБСЛУЖИВАНИЕ ]



Использование API  
должно быть \*\* простым \*\*  
и \*\* быстрым \*\*.





СОВМЕСТИМОСТЬ

[ САМООБСЛУЖИВАНИЕ ]

Использование API должно быть  
как можно более простым.

Подумайте о \*\* опыте  
разработчиков \*\*.

Лучшим решением для проверки  
адекватности с необходимостью  
является \*\* быстрое тестирование  
API \*\*: достаточно нескольких  
минут!

Платформа должна предлагать \*\*  
графический интерфейс \*\* для  
простого тестирования API.



СОВМЕСТИМОСТЬ



[ УПРАВЛЕНИЕ API ]

Использование API  
должно контролироваться  
\*\* и \*\* контролироваться \*\*.





СОВМЕСТИМОСТЬ

[ УПРАВЛЕНИЕ API ]

Внедрите решение управления  
API для управления \*\* квотами \*\*, \*\*  
дресселированием \*\*, \*\*  
аутентификацией \*\* и \*\*  
протоколированием \*\*.

Сбор показателей для управления  
\*\* мониторингом \*\*, \*\* фильтрацией  
\*\* и \*\* отчетностью \*\*.



СОВМЕСТИМОСТЬ



[ ТРЕБОВАНИЯ ]

Задайте \*\* требования \*\*  
для \*\* внешних систем и  
услуг \*\*, встроенных в  
платформу.





## СОВМЕСТИМОСТЬ

### [ ТРЕБОВАНИЯ ]

Требовать \*\* внешние системы \*\*  
для удовлетворения тех же \*\*  
требований \*\* как \*\* внутренние  
системы \*\*.

Внешние системы должны  
публиковать \*\* события \*\* и  
допускать \*\* технический \*\*  
мониторинг.

В случае, когда данные внешних  
систем должны быть  
интегрированы, \*\* полная \*\*  
синхронизация должна быть \*\*  
возможной \*



СОВМЕСТИМОСТЬ



[ МНОГОКВАРТИРНЫЕ  
ДОМА ]

Архитектура должна  
считаться \*\* мульти-  
арендатором \*\*.





СОВМЕСТИМОСТЬ

## [ МНОГОКВАРТИРНЫЕ ДОМА ]

Даже если белая метка не рассматривается в базе, настройте архитектуру с несколькими арендаторами. Ваша \*\* первоначальная \*\* заявка является первой \*\* удержанием \*\*.

Подумайте о \*\* многофункциональном экземпляре \*\* системы с самого начала.





СОВМЕСТИМОСТЬ



[ SETTING ]

Системы должны быть  
конфигурируемы \* \* \*.





## СОВМЕСТИМОСТЬ

[ SETTING ]

\*\* Языки, валюты, бизнес-правила, профили безопасности \*\* должны быть простыми в установке.

Остерегайтесь \*\* гипер-ротации \*\*, это часто бесполезно и \*\* источник затрат \*\*.

\*\* \*\* установка \*\* должна быть масштабируемой \*\* и быстро при необходимости.



СОВМЕСТИМОСТЬ



[ ХАРАКТЕРИСТИКА  
ПЛАНИРОВАНИЯ ]



Создавайте гибкие и  
общие системы, используя  
\*\* функцию flipping \*\*.





СОВМЕСТИМОСТЬ

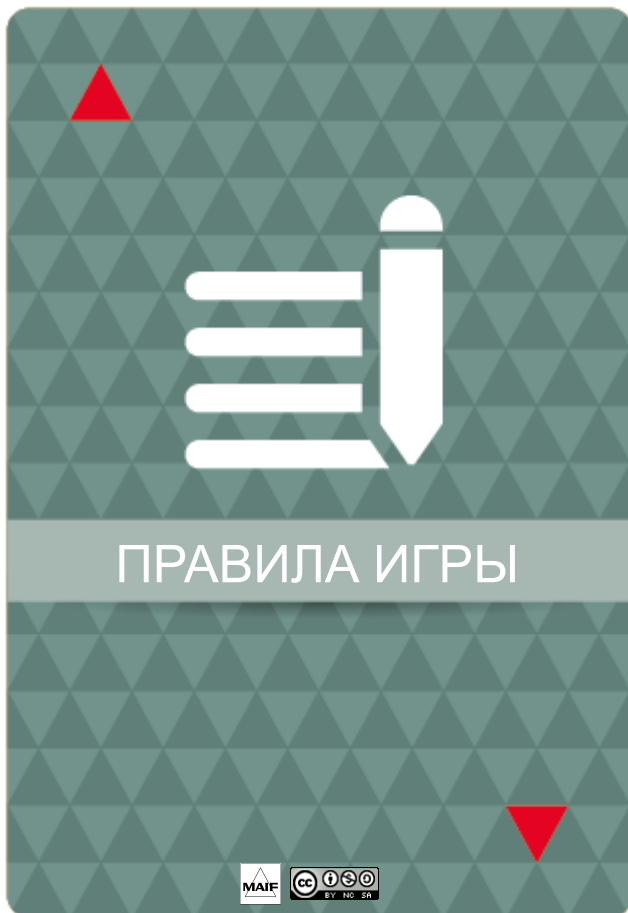
## [ ХАРАКТЕРИСТИКА ПЛАНИРОВАНИЯ ]

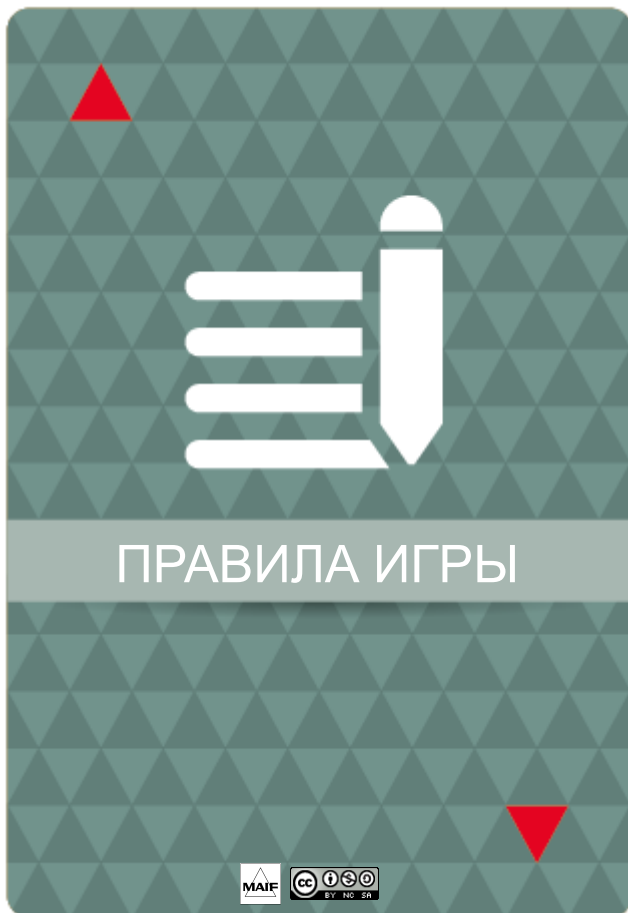
\*\* функция flipping \*\* заключается в разработке приложения в виде набора \*\* функций \*\*, которые могут быть \*\* включены \*\* или \*\* отключены \*\* горячие, \*\* production \*\*.

В приложении \*\* multi-tenant \*\* функция flipping позволяет вам настроить \*\* сторонников.

Le функция flipping \*\* simplifie l'A / В тестирование \*\*.









ПРАВИЛА ИГРЫ

[ ТЕХНИЧЕСКИЕ  
ВЫБОРЫ ]





## ПРАВИЛА ИГРЫ

### [ ТЕХНИЧЕСКИЕ ВЫБОРЫ ]

Функциональная группа должна действовать \*\* ответственно \*\*, чтобы определить варианты, которые влияют на нее исключительно, и варианты, влияющие на организацию.

\*\* выбор \*\*, который \*\* превышает объем \*\* функциональной группы (например, лицензия, несчастный случай, язык программирования), должен быть \*\* подтвержден \*\* организацией или процессом конвергенции сверстников ,







**\*\* Правильный инструмент  
\*\* для \*\* хорошего  
использования \*\* является  
источником экономии.**





## ПРАВИЛА ИГРЫ

### [ Хорошее использование ]

**\*\* Плохой инструмент \*\***,  
наложенный на каждого, является  
**\*\* риском \*\***. **\*\* неправильное**  
**использование \*\*** хорошего  
инструмента может иметь **\*\* очень**  
**\*\* ущерб \*\*** последствия **\*\***.  
Например, опасные методы Agile  
опасны.

**\*\* Инструменты \*\*** должны быть  
поставлены под сомнение **\*\***.

**\*\* Excel \*\*** часто является  
рациональным выбором **\*\***, но это  
не **\*\* инструм**



**\*\* (CRM, ERP, B...**



Привилегия \*\* Сборка \*\*  
для основного бизнеса.

Рассмотрим \*\* Купить \*\*  
для остальных, в каждом  
конкретном случае.







## ПРАВИЛА ИГРЫ

### [ СТРОЙ ВС. КУПИТЬ ]

Чем больше инструмент использует \*\* функцию, отличающую \*\* функцию для организации, тем больше она должна быть построена \*\*. Основной бизнес должен позволять \*\* специфика \*\* и \*\* быстро и часто адаптироваться \*\*. Некоторые \*\* программные пакеты \*\* \*\* иногда адаптируются \*\* к этой потребности.

Для \*\* остальное \*\*: SaaS, Open Source, Build или Ownер должны быть изучены   в каждом случае



ПРАВИЛА ИГРЫ

[ ОТКРЫТЫЙ ИСХОДНЫЙ  
КОД ]

**\*\* Максимально  
используйте Open Source**

**\*\*  
,**

**Альтернативные варианты  
должны поддерживаться.**





## ПРАВИЛА ИГРЫ

### [ ОТКРЫТЫЙ ИСХОДНЫЙ КОД ]

**\*\* Запатентованные решения \*\***  
являются **\*\* риском \*\*** для  
организации, которая должна  
иметь возможность возобновить  
обслуживание, если это  
необходимо.

Существует несколько  
проприетарных инструментов,  
которые не имеют **\*\* альтернатив с**  
открытым исходным кодом **\*\***.

Организация **\*\* извлекает выгоду \*\***  
из **\*\* Сообщения об открытии**



**исходным кодом \*\* и может \*\***

ПРАВИЛА ИГРЫ



[ MICRO-УСЛУГИ ]



Разработка \*\* автономных  
\*\* и \*\* слабо связанных \*\*  
услуг.





## ПРАВИЛА ИГРЫ

### [ MICRO-УСЛУГИ ]

**\*\* слабая связь \*\*** должна быть нормой.

Каждая микросервис имеет **\*\*** четко определенный интерфейс **\*\***.

Этот **\*\* интерфейс \*\*** определяет **\*\*** ссылку **\*\*** между **\*\*** микросервисами **\*\***.

**\*\* Domain Driven Design \*\*** позволяет, особенно с **\*\*** Ограниченными контекстами **\*\***, предвидеть эту проблему.







У каждой службы есть  
своя система хранения  
данных \*\*.





## ПРАВИЛА ИГРЫ

### [ DATA ]

\*\* Хранилище данных \*\* должно быть \*\* соединено \*\* только с \*\* одним микросервисом \*\*.

\*\* Доступ к данным \*\* с одного микросервиса на другой осуществляется \*\* исключительно через его интерфейс \*\*.

Этот дизайн подразумевает \*\* согласованность во времени \*\* по всей платформе. Он должен быть воспринят на всех уровнях \*\*, включая UX.





Каждое микросервис  
должно иметь разумный  
функциональный  
периметр, который \*\*  
«вписывается в голову» \*\*.



## ПРАВИЛА ИГРЫ

### [ ОБЛАСТЬ ПРИМЕНЕНИЯ ]

Микросервис предлагает \*\*  
разумное количество функций \*\*.

\*\* Не стесняйтесь сокращать \*\*  
микросервис, когда он начинает  
расти.

Услуга разумного размера  
позволяет \*\* спокойно рассмотреть  
\*\* переписывание \*\*, если  
возникнет такая необходимость.



ПРАВИЛА ИГРЫ



[ ОТЗЫВЧИВЫЙ ]

**\*\* Reactive Manifesto \*\***  
открывает путь к дизайну  
реактивных архитектур.






## ПРАВИЛА ИГРЫ

[ ОТЗЫВЧИВЫЙ ]

**\*\* Отзывчивое \*\***

программирование фокусируется на потоке данных и распространении изменений. Он основан на схеме «\*\* Наблюдатель», **противоречащей подходу «Итератор \*\*»**, более традиционному.

Реактивный манифест устанавливает основные оси: \*\* доступность \*\* и скорость, \*\* устойчивость к ошибкам, \*\* гибкость \*\*, \*\* эластичность \*\* и \*\* ориентация сообщения \*\*. 



ПРАВИЛА ИГРЫ

[ АСИНХРОННЫЙ  
ПЕРВАЯ ]

**\*\* асинхронные \*\***  
**процессы предпочитают \*\***  
**развязку \*\* и \*\***  
**масштабируемость \*\* в**  
**пользу \*\***  
**производительности \*\*.**





## ПРАВИЛА ИГРЫ

### [ АСИНХРОННЫЙ ПЕРВАЯ ]

Сначала обмен между приложениями должен быть \*\* асинхронным \*\*.

Асинхронные обмены естественно \*\* позволяют \*\* слабую \*\* связь, \*\* изоляцию \*\* и \*\* управление потоком \*\* (\*\* обратное давление \*\*).

\*\* синхронная связь \*\* следует рассматривать только \*\*, когда действие требует этого \*\*.







Информационная система  
должна быть  
ориентирована на \*\*  
события \*\*.





ПРАВИЛА ИГРЫ

[ СОБЫТИЯ ]

Ориентация \*\* \*\* позволяет поддерживать реализацию таких подходов, как \*\* C \*\* ommand \*\* Q \*\* uery \*\* R \*\* Ответственность \*\* S \*\* egregation ( \*\* CQRS \*\* ) и \*\* Event Sourcing \*\*.



ПРАВИЛА ИГРЫ



[ БРОКЕР СООБЩЕНИЯ ]



Привилегия \*\* простого,  
надежного и мощного \*\*  
брокер-сообщения для  
«умной трубы».





## ПРАВИЛА ИГРЫ

### [ БРОКЕР СООБЩЕНИЯ ]

\*\* ESB \*\* показал \*\* пределы \*\*: \*\* масштабируемое обслуживание \*\* является критическим \*\*, как с \*\* технической \*\*, так и с \*\* организационной \*\* точки зрения.

\*\* \*\* Брокер \*\* сообщения, такие как \*\* Kafka \*\* предлагают простые \*\*, \*\* прочные \*\* и \*\* упругие \*\* решения.

\*\* Умные конечные точки \*\* и \*\* Простые трубы \*\* - это архитектура, которая работает в масштабе: это \*\* Интернет \*\*.



**\*\* Полная синхронизация  
\*\* системы должна  
учитываться, как только  
она \*\* разработана \*\*.**



## ПРАВИЛА ИГРЫ

### [ СРОКИ ]

Если \*\* синхронизация \*\* между двумя системами обеспечивается \*\* потоком событий \*\*, общая \*\* пересинхронизация \*\* этих систем должна быть запланирована \*\* во время разработки \*\*.

Автоматический \*\* \*\* \*\* синхронный аудит \*\* (пример: с помощью выборок) позволяет \*\* измерять \*\* и \*\* обнаруживать \*\* любые возможные \*\* ошибки синхронизации \*\*.





ПРАВИЛА ИГРЫ

[ ЦЕНТРАЛИЗАЦИЯ ]

**\*\* Конфигурация \*\* услуг \*\*  
централизована \*\*, их \*\*  
открытие \*\*  
обеспечивается \*\*  
справочником \*\*.**





## ПРАВИЛА ИГРЫ

### [ ЦЕНТРАЛИЗАЦИЯ ]

\*\*\* централизованная \*\*  
\*\* для всех \*\* окружения \*\*.

Центральный справочник \*\*  
обеспечивает \*\* динамическое  
открытие \*\*\* микро-услуг \*\*.

\*\*\* глобальная \*\*  
масштабируемость \*\* зависит от  
этого \*\* каталога \*\*.







ПРАВИЛА ИГРЫ

[ ПЕСКИ ЛОТОК ]

Группы функций  
предоставляют \*\*  
изолированную среду \*\*.





## ПРАВИЛА ИГРЫ

### [ ПЕСКИ ЛОТОК ]

Группы функций поддерживают среду \*\* sandbox \*\* (текущая версия и предстоящий выпуск), чтобы другие \*\* команды расширялись \*\*.

В \*\* некоторых не номинальных \*\* случаях

\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*



ПРАВИЛА ИГРЫ



[ ДИЗАЙН ДЛЯ  
НЕИСПРАВНОСТИ ]



**\*\* Ваша система  
срабатывает! \*\***

Создайте его так, чтобы он  
был терпимым.





ПРАВИЛА ИГРЫ

## [ ДИЗАЙН ДЛЯ НЕИСПРАВНОСТИ ]

Ваша \*\* система не работает , **это неизбежно. Он должен быть разработан для этого** ( Design For Failure \*\*).

Предсказывать \*\* избыточность \*\*  
на всех уровнях: \*\* аппаратное  
обеспечение \*\* (сеть, диск и т. Д.), \*\*  
приложения \*\* (несколько  
экземпляров приложений), \*\*  
географические \*\* зоны, \*\*  
провайдеры \*\* (пример: AWS +  
OVH).





Предоставьте \*\* набор  
инструментов \*\*, не  
налагайте строгие рамки.





## ПРАВИЛА ИГРЫ

[ ToolKits ]

\*\* Внимание к техническим  
компонентам домов и поперечным  
\*\*! Они являются  
ограничительными, дорогими и  
сложными в обслуживании.

\*\* ускорители \*\*, \*\* набор  
инструментов \*\*, \*\* технические  
стеки \*\* могут быть объединены \*\*,  
\*\* бесплатно \*\* Команды функций,  
избегая догматического подхода.





Общественный, частный  
или гибридный, \*\* облако \*\*  
(\* IaaS \* или \* PaaS \*)  
является стандартом для  
производства.





## ПРАВИЛА ИГРЫ

[ CLOUD ]

**\*\* Услуги PaaS \*\***

предпочтительны \*\*, \*\* простые \*\* и быстро масштабируются.

**\*\* Услуги IaaS \*\*** позволяют решать случаи, требующие большей \*\* гибкости \*\*, но требуют более оперативной работы.

Частное облако не является традиционной средой виртуализации, она полагается на \*\* товарное оборудование \*\*.







ПРАВИЛА ИГРЫ

[ ИНФРАСТРУКТУРА ]

Функциональные группы  
не управляют  
инфраструктурой, она \*\*  
предоставляется и  
поддерживается  
организацией \*\*.





## ПРАВИЛА ИГРЫ

### [ ИНФРАСТРУКТУРА ]

Проблемы с инфраструктурой не входят в \*\* Функциональные группы \*\*. Инфраструктура должна быть \*\* предоставлена \*\* и \*\* поддерживается \*\* \*\* кросс-функциональной \*\* услугой.





**\*\* Контейнеры \*\***  
обеспечивают гибкость,  
необходимую для  
гетерогенной оснастки.





## ПРАВИЛА ИГРЫ

### [ КОНТЕЙНЕРЫ ]

Контейнеры обеспечивают \*\*  
гибкость \*\*, необходимую  
функциональным группам для  
включения \*\* гетерогенной  
оснастки \*\* в \*\* однородном  
контексте \*\*.



Использование \*\*  
контейнеров \*\* позволяет  
преодолеть проблемы \*\*  
технических условий \*\*.



## ПРАВИЛА ИГРЫ

### [ ОКРУЖЕНИЕ ]

Контейнеры (пример: \*\* Docker \*\*) позволяют \*\* освободиться \*\* от различий в окружающей среде.

Процесс \*\* развертывания \*\* должен быть \*\* агностиком \*\* для среды.

\*\* Некоторые компоненты \*\*, такие как базы данных, не должны развертываться в контейнерах. Их развертывание по-прежнему автоматизировано.



Меры должны быть \*\*  
централизованы \*\* и \*\*  
доступны \*\* всем.



## ПРАВИЛА ИГРЫ

### [ метрическая ]

\*\* \*\* \*\* \*\* \*\* доступны \*\* для всех с разным уровнем детализации: подробное представление для соответствующей функции команды, скопления для других членов организации.

Доступ к \*\* метрикам не подразумевает доступ к данным о единице \*\*, он должен контролироваться для обеспечения конфиденциальности.

\*\* Все среды



уты.



\*\* качество программного  
обеспечения \*\* является \*\*  
ключевым фактором \*\*.



## ПРАВИЛА ИГРЫ

### [ КАЧЕСТВО ]

**\*\* обзоры кода \*\*** являются **\*\* систематическими \*\***. Они проводятся членами функциональной группы или другими членами организации в рамках **\*\* Постоянного совершенствования \*\***.

Это **\*\* вы не проверяетесь, а ваш код \*\***: «Вы не ваш код!».

**\*\* qualimetry \*\*** может быть частично автоматизирован, но ничто не сравнится с **\*\* «новым глазом» \*\***.





ПРАВИЛА ИГРЫ

[ АВТОМАТИЗАЦИЯ ]

\*\* автоматическое  
тестирование \*\* является  
необратимым  
предварительным  
условием для  
непрерывного  
развертывания.





## ПРАВИЛА ИГРЫ

### [ АВТОМАТИЗАЦИЯ ]

Автоматизированное \*\*  
тестирование \*\* гарантирует \*\*  
качество \*\* продукта \*\* со  
временем \*\*.

Это \*\* предварительное условие \*\*  
для непрерывного развертывания,  
оно позволяет \*\* \*\* изменения \*\* и  
\*\* частое развертывание \*\*.

\*\* Производственный свиток \*\*  
становится \*\* анекдотическим \*\*  
событием!





**\*\* Тесты на всех уровнях \*\*:**   
единица, интеграция,  
функциональность,  
устойчивость,  
производительность.





ПРАВИЛА ИГРЫ

[ УРОВНИ ТЕСТА ]

\*\* \*\* тесты \*\* подходят для \*\*  
разработки \*\*.

\*\* выступление \*\* тест измерение  
работоспособность \*\* со временем  
\*\*.

\*\* Устойчивость \*\* тесты помогают  
прогнозировать \*\* сбои \*\*.



**\*\* Обложка \*\* является  
основным объективным  
показателем качества  
теста.**



## ПРАВИЛА ИГРЫ

### [ ПОКРОВ ]

**\*\* Кодирование \*\*** в тестах является **\*\* хорошим \*\*** метрикой качества кода.

Это **\*\* необходимое условие \*\***, но **\*\* недостаточно \*\***, охват **\*\* плохой** тест-теста может быть высоким, не гарантируя хорошего качества кода.







ПРАВИЛА ИГРЫ

[ БЕЗОПАСНОСТЬ ]

\*\* security \*\* - это \*\*  
процесс \*\*, он не должен  
рассматриваться в ответ  
на проблемы.





## ПРАВИЛА ИГРЫ

### [ БЕЗОПАСНОСТЬ ]

\*\* эксперты по безопасности \*\*  
могут быть \*\* интегрированы \*\*  
непосредственно в группы  
функций \*\* при необходимости \*\*.

\*\* эксперты по безопасности \*\*  
доступны в организации для \*\*  
аудита \*\*, \*\* осведомленности \*\* и  
\*\* вперед \*\*.

