

НЕ ТОЛЬКО DEVELOPER EXPERIENCE

Что на самом деле нужно для успешной
внутренней платформы разработки?



О спикере



Зимонин
Алексей
технический
лидер Э42

Чем занимаюсь

Веду консалтинговые проекты, помогаю строить платформы и развивать процессы

Опыт

- С 2008 в ИТ
 - С 2016 развивал highload-инфраструктуру в телекоме
 - С 2018 строил инфраструктурную платформу
 - С 2022 присоединился к команде «Экспресс 42» в качестве техлида
-

azimonin@express42.com
tg: @zlyoha

Экспресс 42

«Экспресс 42» — это российский центр экспертизы по DevOps с 2012 года

Мы предлагаем широкий спектр услуг для того, чтобы вывести поставку цифровых продуктов на новый уровень и извлечь максимум ценности из Cloud Native-технологий

13

лет в DevOps
и SRE

100+

реализованных проектов

State of DevOps Russia

Создатели и авторы ежегодного исследования о состоянии DevOps в России



2023 год — объединение
с АО «Флант»



ФСТЭК



Аккредитация
Минцифры



Реестр
российского ПО

ОРГАНИЗАЦИОННЫЕ МОМЕНТЫ

Длительность вебинара — **60 минут** вместе с Q&A

Запись будет доступна **публично** на платформе **МТС Линк** и **Rutube-канале «Фланта»**

В конце вебинара будет **блок вопросов и ответов**

Оставляйте ваши вопросы на **специальной вкладке со знаком «?»**

Активно общайтесь в чате, **мы за живую дискуссию :)**

ВВЕДЕНИЕ



ЧТО ТАКОЕ ПЛАТФОРМА?



Что?



Платформа

— продукт, дающий
согласованные возможности

* Платформа — интегрированный продукт, предоставляющий согласованный набор возможностей для решения задач своих пользователей независимо от их сферы применения — будь то разработка программного обеспечения, управление инфраструктурой, бизнес-процессы или предоставление сервисов. Она обеспечивает стандартизированный и удобный пользовательский опыт через интерфейсы, такие как API, веб-порталы или инструменты самообслуживания. Платформа выступает как универсальный слой, упрощая взаимодействие с технологическими и бизнес-решениями, абстрагируя сложные процессы и стандартизируя подходы к выполнению задач, что делает её применимой в различных доменах.

Что?



Платформа

— продукт, дающий
согласованные возможности



Внутренняя платформа разработки (IDP)

— платформа для разработчиков
внутри компании

* Платформа — интегрированный продукт, предоставляющий согласованный набор возможностей для решения задач своих пользователей независимо от их сферы применения — будь то разработка программного обеспечения, управление инфраструктурой, бизнес-процессы или предоставление сервисов. Она обеспечивает стандартизированный и удобный пользовательский опыт через интерфейсы, такие как API, веб-порталы или инструменты самообслуживания. Платформа выступает как универсальный слой, упрощая взаимодействие с технологическими и бизнес-решениями, абстрагируя сложные процессы и стандартизируя подходы к выполнению задач, что делает её применимой в различных доменах.

Где?

Сегодня не будем отвечать на этот вопрос

Когда?



Масштабирование разработки

Несколько команд пользуются
одним стеком



Требования, политики, регламенты

Процесс усложняется
и его сложно соблюдать вручную



Сложность инфраструктуры

Для запуска Hello world
тратится больше суток

Когда?

✓ Заполнение опросника (DORA)

2 Заполнение опросника (практики DevOps)

3 Результат

Эффективность поставки программного обеспечения

Оцените эффективность работы вашей команды по поставке программного обеспечения менее чем за минуту! Сравните ее с остальными в отрасли в России, ответив на несколько простых вопросов.

Срок поставки

Сколько в среднем проходит времени между окончанием разработки кода (последний коммит или оформленный Merge Request) и его развертыванием на продуктовом окружении?

- Более 6 месяцев
- От 3 до 6 месяцев
- От месяца до 3 месяцев
- От 2 недель до месяца
- От недели до 2 недель
- От дня до недели
- Меньше дня
- Меньше часа

Частота развертывания

Как часто происходит развертывание новой версии продукта на продуктовое окружение? Считая только плановые изменения, исключая хотфиксы и реакцию на инциденты.

- Реже чем раз в 6 месяцев
- От раза в 3 месяца до раза в 6 месяцев

Кто?

В главных ролях



Руководитель
платформы

Product owner,
руководитель инфраструктуры



Платформенный
инженер

DevOps-инженер,
SRE-специалист



Разработчик
Тестирующий

Цели внедрения IDP



Скорость поставки ПО

Снижение времени на проверку продуктовых гипотез.



Снижение стоимости владения

Управление коммунальными ресурсами, мультитенантность, уменьшение дублирования инфраструктуры.



Гибкость

Платформа выступает рычагом для быстрой адаптации команд разработки к новым технологиям.



Масштабируемость

Инструмент для распространения практик и культуры разработки при масштабировании команд. Возможности для ускорения онбординга.



Доступность

Предоставление «из коробки» инструментов для повышения доступности приложений.



Качество обслуживания

Единое окно для автоматизации и сопровождения процессов разработки.

Цели внедрения IDP



Руководитель платформы

- Снижение TTM
- Снижение TCO



Платформенный инженер

- Масштабируемость технологий
- Внедрение лучших практик
- Стабилизация инфраструктуры



Разработчик

- Удобный и быстрый онбординг
- Возможность больше кодить



Ценность внутренней платформы разработки (IDP)

1 Индивидуальная производительность

В компаниях, использующих IDP, производительность труда разработчиков выше **на 8 %**

2 Командная производительность

В компаниях, использующих IDP, производительность труда команд разработки выше **на 10 %**

3 Скорость поставки ПО

В компаниях, использующих IDP, скорость поставки ПО выше **на 6 %**



Почему проекты по внедрению IDP не достигают целей?



Сложность

Обеспечение производственной масштабируемости оказывается намного сложнее, чем предполагалось на начальных этапах создания IDP.



Непрозрачные роли и зоны ответственности

Успех IDP в организации требует высоких компетенций платформенных команд для того, чтобы абстрагировать сложность работы с инфраструктурой от продуктовых команд разработки.



Отсутствие продуктового фокуса

Организации воспринимают IDP в виде разовой инициативы, реализованной в виде набора скриптов, а не как живой и развивающийся продукт.



Отсутствие адаптивности

Технологии быстро эволюционируют, неспособность быстро интегрировать инновации в IDP быстро приводит к потере ей релевантности.



Затраты

IDP может отнимать до 25 % ресурсов DevOps- и платформенных команд. Дает ли IDP собственной разработке дифференцирующие способности бизнесу?



Отсутствие наблюдаемости

Для успеха IDP должна иметь навыки глубокой наблюдаемости, контроля и надзора за приложениями и их зависимостями после развёртывания.

КАК РЕАЛИЗОВАТЬ ПЛАТФОРМУ?



Варианты реализации

Internal Development Platform



Build



Customize



Buy

Варианты реализации

Internal Development Platform



Build

Portal + Backend



Customize



Buy

Варианты реализации

Internal Development Platform



Build

Portal + Backend



Customize



Buy

On-Prem | PaaS

Варианты реализации

Internal Development Platform



Build

Portal + Backend



Customize

OSS | Vendor



Buy

On-Prem | PaaS

BUILD



Время на внедрение

Долго. Включает проектирование, разработку функциональных возможностей с нуля. Сильно зависит от зрелости платформенной команды.



Стоимость

Низкие вложения на старте при реализации существующей командой. Рост затрат по мере развития продукта.



Гибкость

Высокая. Нет технических ограничений.



Масштабируемость

В зависимости от глубины проработки архитектуры на старте проекта и согласованности с потенциалом роста количества разработки.



Риски поддержки

Высокая компетенция внутри на старте, по умолчанию нет компетенций вне компании, что повышает bus factor.



Контроль

Полный контроль над архитектурой решения, безопасностью, данными, зависимостями.

BUY



Время на внедрение

Быстрое. Особенно для небольших команд. Требуется настройка и адаптация процессов под продукт.



Стоимость

Высокая стоимость на старте для on-prem. Снижение затрат и повышение окупаемости на длинной дистанции.



Гибкость

Варьируется в зависимости от базового компонента: OSS или коммерческий продукт.



Масштабируемость

Высокая масштабируемость, особенно в облачных решениях. Поставщики, как правило, оптимизируют масштабирование под массовые сценарии.



Риски поддержки

Минимальные риски — ответственность за стабильность и обновления несёт поставщик. SLA и техподдержка закрывают большинство проблем.



Контроль

Низкий уровень контроля над архитектурой решения, безопасностью, данными, зависимостями. Ограниченная прозрачность и зависимость от вендора. Влияние на приоритет разработки зависит от условий договора.

CUSTOMIZE



Время на внедрение

Быстрее, чем build, но при проведении MVP требует времени на адаптацию под процессы.



Масштабируемость

Варьируется в зависимости от базового компонента: OSS или коммерческий продукт.



Стоимость

Зависит от выбранной платформы (Open Source или коммерческой).
Требуются инвестиции в доработку и настройку под свои процессы.



Риски поддержки

Варьируется в зависимости от базового компонента: OSS или коммерческий продукт.



Гибкость

Адаптация платформы ограничена доступными настройками и функционалом.
Изменения под свои процессы часто невозможны.



Контроль

Чаще высокий.

Варианты реализации

Вариант	Скорость внедрения	Стоимость запуска	Гибкость	Контроль	Масштабируемость	Риски поддержки
Build	Низкая 	Низкая 	Высокая 	Высокий 	Нейтральная 	Высокий 
Custom OSS	Нейтральная 	Низкая/ нейтральная 	Высокая 	Высокий 	Нейтральная 	Нейтральный 
Custom Vendor	Нейтральная 	Нейтральная/ высокая 	Высокая 	Нейтральный 	Высокая 	Низкий 
Buy	Высокая 	Высокая 	Низкая 	Низкий 	Высокая 	Низкий 

КАКИЕ СУЩЕСТВУЮТ ПРИЗНАКИ КАЧЕСТВА ПЛАТФОРМЫ?



Рассмотрим 3 модели зрелости платформы

1

CNCF

Cloud Native Computing Foundation

2

7C

Gregor Hohpe

3

Express Framework

PaaS тоже сюда

CNCF

Ключевые свойства платформы
по версии фонда CNCF



CNCF: 7 атрибутов платформы



Платформа как продукт

Должна разрабатываться и развиваться с учётом потребностей пользователей.



Пользовательский опыт

Должна предоставлять свои возможности через единообразные интерфейсы и фокусироваться на пользовательском опыте.



Документация и онбординг

Предоставляет документацию и примеры.



Самообслуживание

Пользователи должны иметь возможность самостоятельно и автоматически запрашивать и получать необходимые им функции.



Снижение когнитивной нагрузки для пользователей

Должна инкапсулировать детали реализации и скрывать любую сложность, обусловленную её архитектурой.



Опциональная и комбинируемая

Должна быть модульной и позволять продуктовым командам использовать только часть её возможностей.



Безопасная по умолчанию

Должна быть безопасной «из коробки» и позволять проверять систему на соответствие политикам и стандартам, принятым в организации.

CNCF

Платформа как продукт



Руководитель платформы →
владелец продукта



Платформа без пользователей —
деньги на ветер



Гарантии от владельца продукта

CNCF

Платформа как продукт



Руководитель платформы →
владелец продукта



Платформа без пользователей —
деньги на ветер



Гарантии от владельца продукта



CNCF

Платформа как продукт



Руководитель платформы →
владелец продукта



Платформа без пользователей —
деньги на ветер



Гарантии от владельца продукта



CNCF

Платформа как продукт

➤ Успешные примеры

- метрика скорости онбординга новых пользователей и запуска продуктов предоставляет данные об эффекте от внедрения IDP
- ретроспективно анализируем данные о наиболее востребованном функционале платформы и приоритезируем бэклог развития в соответствии с релевантными запросами пользователей



CNCF

Платформа как продукт

➤ Спорные примеры

- выбор недостаточных метрик для оценки качества продукта (например, измеряем доступность, но без оценки задержки со стороны конечного пользователя)
- пользователи не видят прозрачно информацию о том, какие возможности появятся в платформе, из-за отсутствия дорожной карты не видят стимула к её использованию



CNCF

Безопасная по умолчанию



Снижение рисков ИБ



Централизованная конфигурация
политик



Не нужно отдельное согласование



CNCF

Безопасная по умолчанию

➤ Успешные примеры

- централизованная конфигурация Open Policy Agent (OPA) в Kubernetes
- управление сетевыми политиками, возможность управления через IaC с согласованием ИБ
- унифицированный подход к хранению секретов (реализация подходов secret management) согласно ролевой модели



CNCF

Безопасная по умолчанию

➤ Спорные примеры

- дублирование инструментов согласования в платформе и в смежных с платформой подразделениях (подтверждение релизов, согласования с ИБ);
- снижение бдительности разработки при буквальной трактовке данной практики



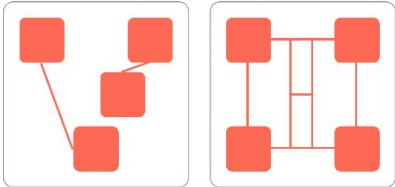
7C

Модель Gregor Hohpe — автора книги «Platform Strategy: Innovation through Harmonization»

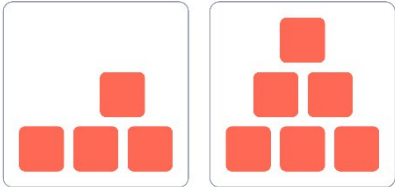


Атрибуты платформы по модели 7С

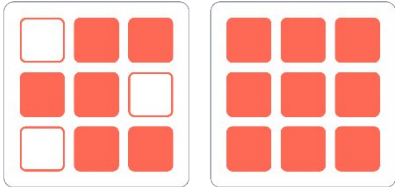
Связность



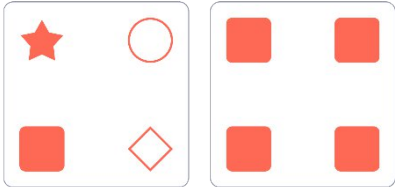
Самодостаточность



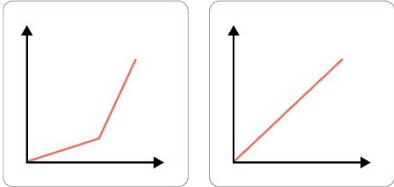
Синергия



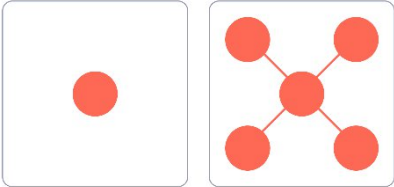
Согласованность



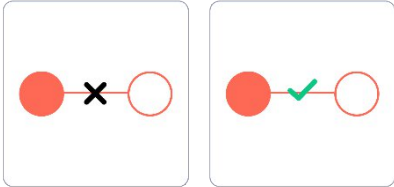
Соразмерная ценность



Связанность

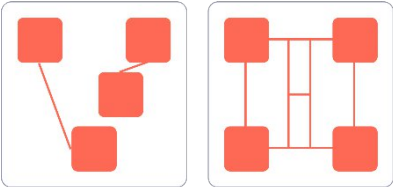


Свободный переход

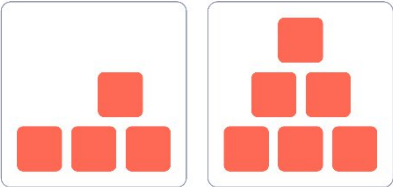


Атрибуты платформы по модели 7C. English

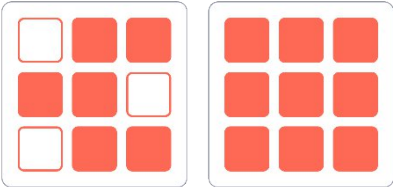
Cohesion



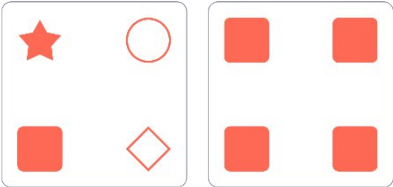
Completeness



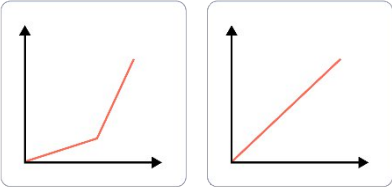
Closure



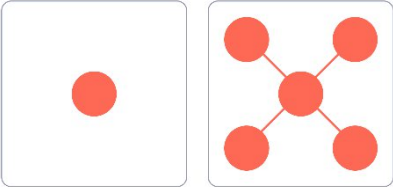
Consistency



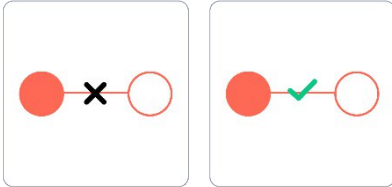
Commensurate Value



Connectedness

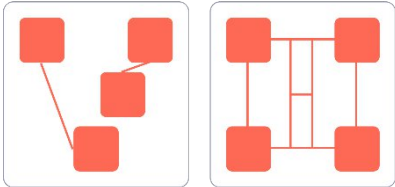


Captivity



Атрибуты платформы по модели 7С

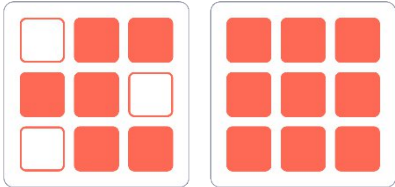
Связность



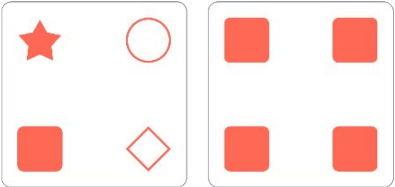
Самодостаточность



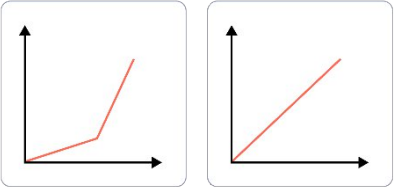
Синергия



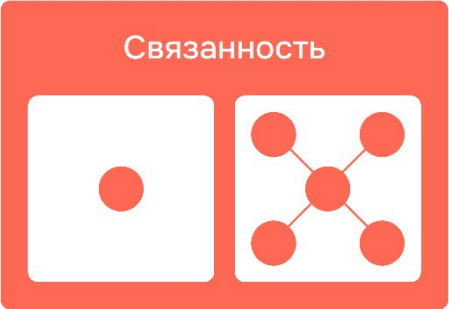
Согласованность



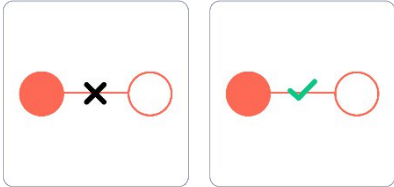
Соразмерная ценность



Связанность



Свободный переход



7C

СВЯЗАННОСТЬ (Connectedness)



Интегрируемся в существующие инфраструктуру и процессы

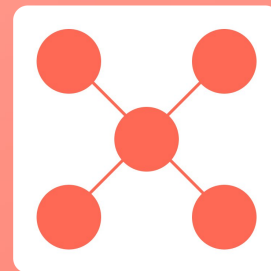
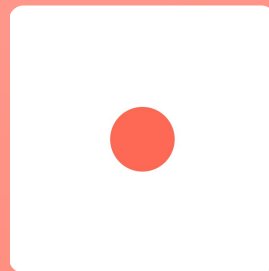


Переиспользуем готовые решения внутри компании



При использовании платформы испытываем знакомый опыт

Связанность



7C

СВЯЗАННОСТЬ (Connectedness)

➤ Успешные примеры

- платформа умеет общаться по стандартным протоколам (OIDC, OpenTelemetry)
- использование единого SSO
- применение существующих инструментов observability (логирования, мониторинга, алертинга, трейсинга)



7C

СВЯЗАННОСТЬ (Connectedness)

➤ Спорные примеры

- высокие трудозатраты на поддержку проприетарного или малораспространённого протокола (Zabbix vs Prometheus)
- добавление нецелевых интеграций в платформу (требовалось одной команде — сделали для всех и поддерживаем)



7C

Самодостаточность (Completeness)



Платформа покрывает весь цикл разработки (SDLC)

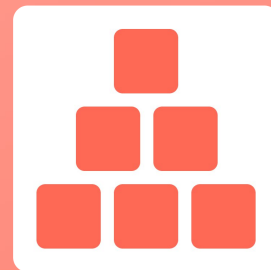
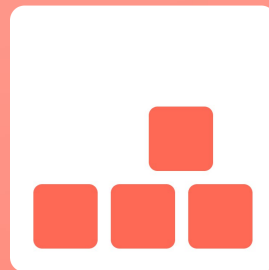


Возможность замены отдельных компонент платформы



Возможность самому обучаться и использовать сервисы платформы в режиме самообслуживания

Самодостаточность



7С

Самодостаточность (Completeness)

➤ Успешные примеры

- портал самообслуживания, который даёт возможность по умолчанию пользоваться платформой без вовлечения команды сопровождения
- база знаний платформы в едином пространстве, дающая готовые ответы на популярные вопросы
- описан публичный кейс golden path, покрывающий большинство кейсов применения платформы



7C

Самодостаточность (Completeness)

➤ Спорные примеры

- отсутствуют триггеры для актуализации базы знаний
- фокусировка на UI без возможности применения средств автоматизации со стороны команды разработки (CLI, Terraform)

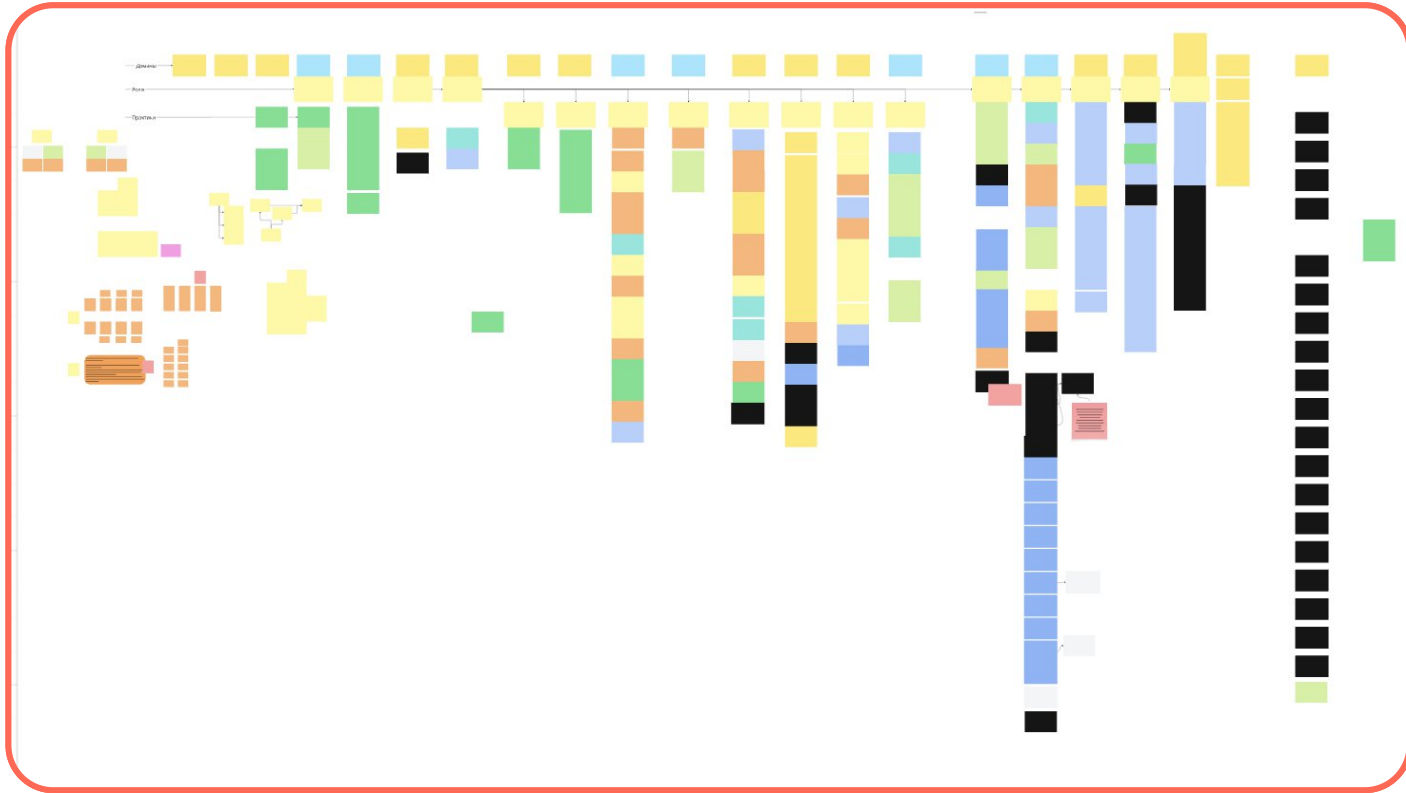


EXPRESS FRAMEWORK

Авторская рамочная модель
и методология поставки «Экспресс 42»






Признаки платформы по модели Express Framework



Express Framework

Центры компетенций

-  Распространение проверенных подходов внутри компании
-  Платформа объединяет уже имеющиеся знания ЦК
-  Разработчик может делиться своей экспертизой через IDP



Express Framework

Центры компетенций

➤ Успешные примеры

- ЦК стандартизируют процессы, платформа их имплементирует в сценариях платформы
- ЦК разработки создают и сопровождают шаблоны репозитория для масштабирования практики inner source и распространения проверенного набора библиотек для подключения к observability-системам новых продуктов.



Express Framework

Центры компетенций

➤ Спорные примеры

- сложные согласования
- уменьшение гибкости применения изменений
- отсутствие возможности отойти от стандартного пути для тестирования новых технологий



Express Framework

Dogfooding



Возможность быстрого тестирования новых функций



Возможность получить качественную обратную связь



Более стабильный продукт



Express Framework

Dogfooding

➤ Успешные примеры

- Product Manager может оперативно проводить CustDev на внутренней команде
- команда платформенной разработки может подсветить качественное решение первопричины проблемы, т. к. знает продукт на высоком уровне



Express Framework

Dogfooding

➤ Спорные примеры

- «замыливание» взгляда
- страх предоставить развивающую обратную связь близкой команде
- нехватка мотивации для платформенной разработки предоставлять обратную связь владельцу продукта



Какие существуют признаки качества платформы?

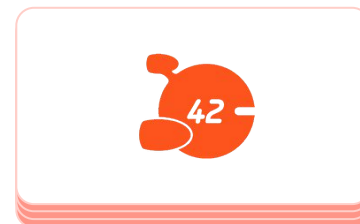
CNCF



7C



Express
Framework



ПОДВЕДЁМ ИТОГИ



Мы сегодня ответили на вопросы

Что? такое платформа,
и IDP в частности



Мы сегодня ответили на вопросы

Что? такое платформа,
и IDP в частности



Где?



Мы сегодня ответили на вопросы



Что? такое платформа,
и IDP в частности



Где?



Когда?



Мы сегодня ответили на вопросы

Что? такое платформа,
и IDP в частности



Кто?



Где?



Когда?



Мы сегодня ответили на вопросы

Что? такое платформа,
и IDP в частности



Кто?



Где?



Как?



Когда?



Мы сегодня ответили на вопросы

Что? такое платформа вообще
и IDP в частности



Кто?



Где?



Как?



Когда?



Какая?



ЧТО ДАЛЬШЕ?



Что делать дальше?

- Не могу определиться
 - пройдите DORA Quick Check
 - пишите нам на input@express42.com
- Планирую внедрить
 - поможем с проектированием
 - поможем с внедрением
- Хочу развить свой IDP
 - разработаем дорожную карту



Полезные материалы



Путеводитель по Platform Engineering [от «Экспресс 42»](#)

Перейти 



DORA Quick Check


Перейти 



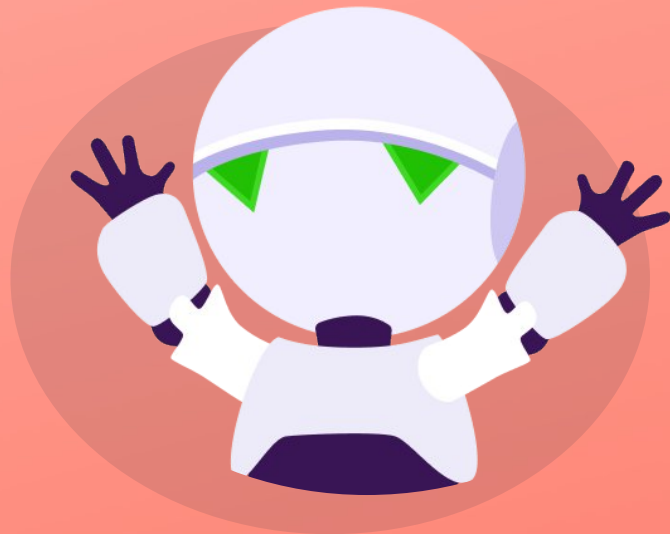
СПАСИБО ЗА ВНИМАНИЕ!

Готовы ответить
на ваши вопросы!

 input@express42.com

 +7 (495) 088-42-84

 express42.com



Полезные материалы



[Путеводитель по зрелости, метрикам и лучшим практикам Platform Engineering от «Экспресс 42»](#)



[DORA Quick Check](#)



Q&A



Варианты реализации

Internal Development Platform



Build

Portal + Backend



Customize

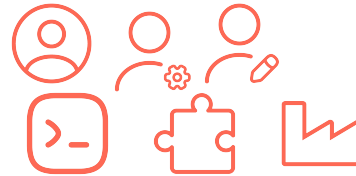
Vendor | PaaS



Buy

Мы сегодня ответили на вопросы

- Что? такое платформа, и IDP в частности (картинка IDP)
- Где? (можно две картинка on-prem и облако)
- Когда? (3 иконки из 10 слайда)
- Кто? (3 иконки по ролям)
- Как? (3 иконки вариантов)
- Какая?



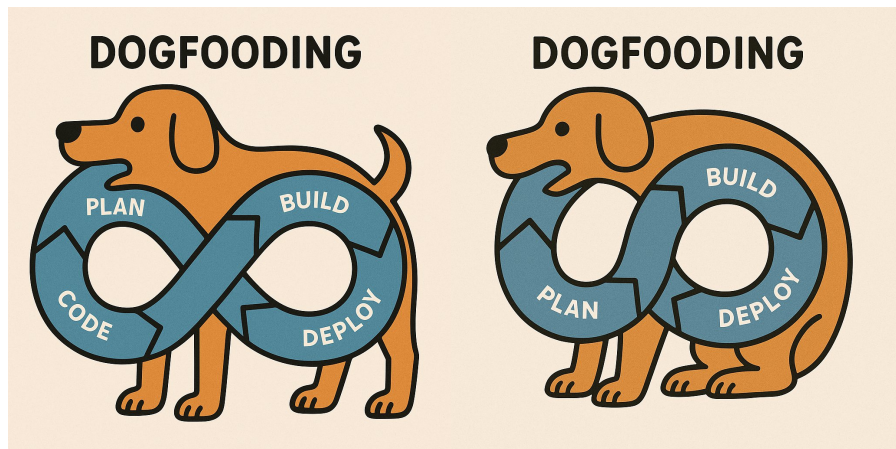
ПОСЛЕДНИЙ СЛАЙД
ВЕБИНАРА, ДАЛЕЕ
ШАБЛОНЫ (которые не
трогаю)



Express Framework

Dogfooding

Нужна запоминающаяся картинка



- Простейшая платформа – набор документации
- Метрика скорости онбординга новых пользователей и продуктов полезна

➤ Успешные примеры

- Стандартизируют процессы
- Разрабатывают и сопровождают шаблоны

➤ Спорные примеры:

- в проектировании.
- в менеджменте проектных работ.

SUMMARY

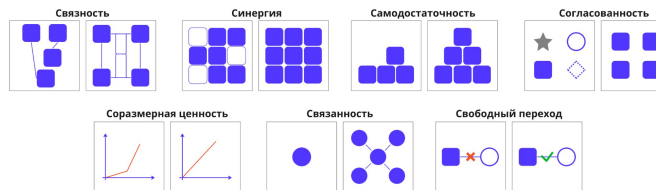
CNCF. 7 атрибутов платформы

7C

Express Framework



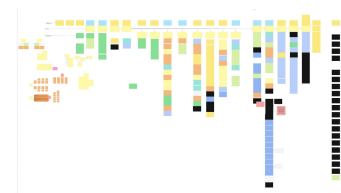
**CLOUD NATIVE
COMPUTING FOUNDATION**



PLATFORM STR
INNOVATION THROUGH HARM

Gre
An Archi

With Contributions by Michele Danielli
and Jean-Francois Landreau



CNCF

Безопасная по умолчанию

- Простейшая платформа – набор документации
- Метрика скорости онбординга новых пользователей и продуктов полезна

➤ Успешные примеры

- база знаний платформы в едином пространстве
- ретроспективно анализируем время на запуск новых продуктов через платформу

➤ Спорные примеры

- в проектировании
- в менеджменте проектных работ

CNCF

Документация и онбординг. ВАРИАНТ 2

- Дает данные для анализа времени на запуск продуктов / скорость проверки гипотез
- Продвигает платформу, меньше однотипных запросов
- Упрощает жизнь новым сотрудникам. Единая точка правды по процессам разработки

➤ Успешные примеры

- база знаний платформы в едином пространстве
- ретроспективно анализируем время на запуск новых продуктов через платформу

➤ Спорные примеры:

- перебор с документацией, которая быстро будет устаревать
- мало новых команд/продуктов

CNCF

Документация и онбординг. ВАРИАНТ 1

- Простейшая платформа – набор документации
- Метрика скорости онбординга новых пользователей и продуктов полезна

➤ Успешные примеры

- база знаний платформы в едином пространстве
- ретроспективно анализируем время на запуск новых продуктов через платформу

➤ Спорные примеры

- в проектировании.
- в менеджменте проектных работ.

Express Framework

Центры компетенций

Не знаю что сюда написать

- Простейшая платформа – набор документации
- Метрика скорости онбординга новых пользователей и продуктов полезна

- **Успешные примеры**
 - Стандартизируют процессы
 - Разрабатывают и сопровождают шаблоны

- **Спорные примеры:**
 - в проектировании.
 - в менеджменте проектных работ.

В главных ролях

Предпосылки использования Kubernetes



Руководитель
платформы

Product owner

руководитель инфраструктуры, можем к
вариациям ролей



Платформенный
инженер

DevOps инженер
SRE-специалист



Разработчик

Тестировщик

CNCF

Документация и онбординг. ВАРИАНТ 2

- Дает данные для анализа времени на запуск продуктов / скорость проверки гипотез
- Продвигает платформу, меньше однотипных запросов
- Упрощает жизнь новым сотрудникам. Единая точка правды по процессам разработки

Возможность добавить картинку

CNCF

Документация и онбординг. ВАРИАНТ 2

➤ Успешные примеры

- база знаний платформы в едином пространстве
- ретроспективно анализируем время на запуск новых продуктов через платформу



CNCF

Документация и онбординг. ВАРИАНТ 2

➤ Спорные примеры

- перебор с документацией, которая быстро будет устаревать
- мало новых команд/продуктов



Слайд с кьюаром на чек лист

На дора квик чек

Ценность внутренней платформы разработки (IDP)

Успешная, клиентоцентричная IDP, положительно влияет на скорость разработки ПО

1

Индивидуальная производительность

В компаниях, использующих IDP, производительность труда разработчиков выше на 8%

2

Командная производительность

В компаниях, использующих IDP, производительность труда команд разработки выше на 10%

3

Скорость поставки ПО

В компаниях, использующих IDP, скорость поставки ПО выше на 6%

CNCF: 7 атрибутов платформы



Платформа как продукт

Должна разрабатываться и развиваться с учётом потребностей пользователей



Пользовательский опыт

Должна предоставлять свои возможности через единообразные интерфейсы и фокусироваться на пользовательском опыте



Документация и онбординг

Предоставляет документацию и примеры



Самообслуживание

Пользователи должны иметь возможность самостоятельно и автоматически запрашивать и получать необходимые им функции



Снижение когнитивной нагрузки для пользователей

Должна инкапсулировать детали реализации и скрывать любую сложность, обусловленную её архитектурой.



Опциональная и комбинируемая

Должна быть модульной и позволять продуктовым командам использовать только часть её возможностей



Безопасная по умолчанию

Должна быть безопасной «из коробки» и позволять проверять систему на соответствие политикам и стандартам, принятым в организации

Если вы только планируете внедрять
платформу или хотите разработать
дорожную карту развития существующей
платформы – мы готовы делиться
экспертизой

Не можете определиться

– проходите **DORA Quick Check**

Хотите спроектировать или внедрить

– приходите к нам за **экспертизой**

Хотите разработать дорожную карту развития –

приходите к нам за **экспертизой**

Не могу определиться

- пройдите DORA quick check
- пишите нам на input@express42.com

