Deckhouse
Development Platform

Internal Development Platform для зрелой разработки

Опыт с IDP от Deckhouse

Deckhouse
Development Platform

Спикеры

Алексей Зимонин

Старший техлид по практикам DevOps,
«Экспресс 42»

alexey.zimonin@flant.com

Никита Вельгин

Технический менеджер продукта,
Deckhouse Development Platform

nikita.velgin@flant.com

О компании «Флант»

Deckhouse
ФЛАНТ

17+
лет опыта
в Open Source

С 2017
года используем
Kubernetes в production

№1
контрибьютор в проекты
CNCF из России

500+
сотрудников

>260
компаний-пользователей

В топе
вендоров ИТ-решений для банков *
и промышленности **

Реестр
российского ПО

Лицензии и сертификат
ФСТЭК России

АРПП «Отечественный
софт»

* Рейтинг «Крупнейшие ИТ-вендоры в банках», TAdviser, 2024
** Рейтинг «Крупнейшие ИТ-вендоры в промышленности», TAdviser, 2024

ФЛАНТ

Синергия опыта вендора, интегратора, сервисной и консалтинговой компании

Deckhouse

Deckhouse — продуктовое подразделение, разработчик продуктов для построения надёжной enterprise-инфраструктуры

DaaS

DaaS — комплексное DevOps-сопровождение инфраструктуры в режиме 24/7 силами выделенной DevOps-команды

Express 42

«Экспресс 42» — DevOps-консалтинг. От анализа узких мест в ИТ-процессах до создания роадмапа изменения ИТ для реализации цифровой трансформации

Когда одного Kubernetes становится недостаточно

для роста зрелости процессов разработки

Deckhouse
Development Platform

Приоритеты в разработке

Менеджмент

- Стандартизация
- Видимость
- Контроль
- Эффективность

Команды разработки

- Скорость
- Стабильность
- Эффективность
- Коммуникация
- Состязание с ИИ (?)

Инфраструктурные команды

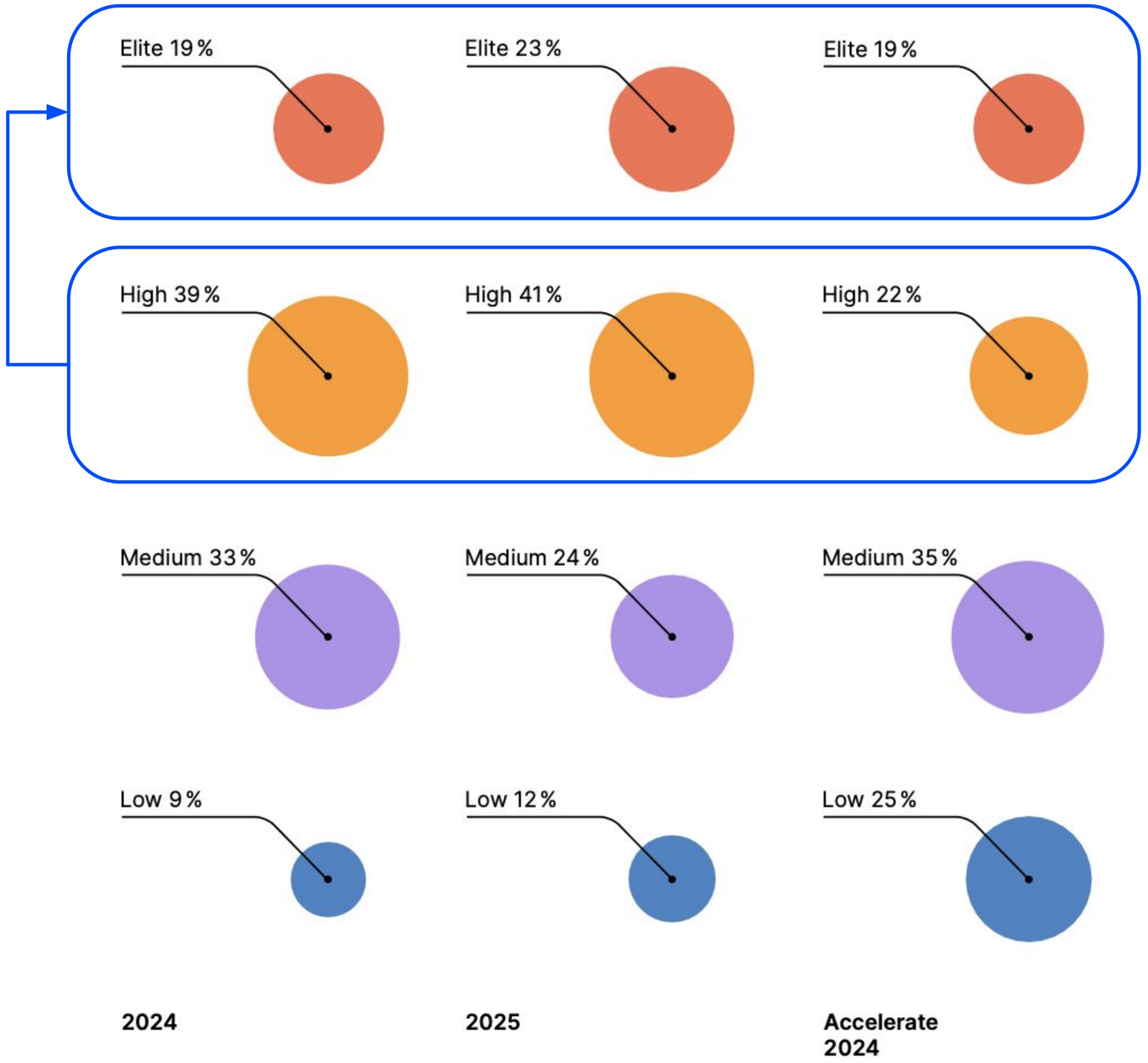
- Доступность
- Стандартизация
- Снижение использования ресурсов

Профили эффективности

как индикаторы потенциала команды для достижения целей

Как перейти из High в Elite?

- Нет универсального рецепта по переходу в Elite
- У нас уже есть Kubernetes, его не хватит?



Почему Kubernetes недостаточно

Скорость поставки ПО

Зависимость от качества
пайплайна сборки и поставки, уровня
реализации практик тестирования

Масштабируемость

Каждая команда может поставлять по-своему,
при этом при росте количества команд
стандарты развиваются независимо

Снижение стоимости владения

Управление ресурсами
в пределах одной платформы

Доступность

Прозрачность по метрикам только
для микросервисной полезной нагрузки

Качество обслуживания

Хорошей технологии нужны хорошие
автоматизированные процессы

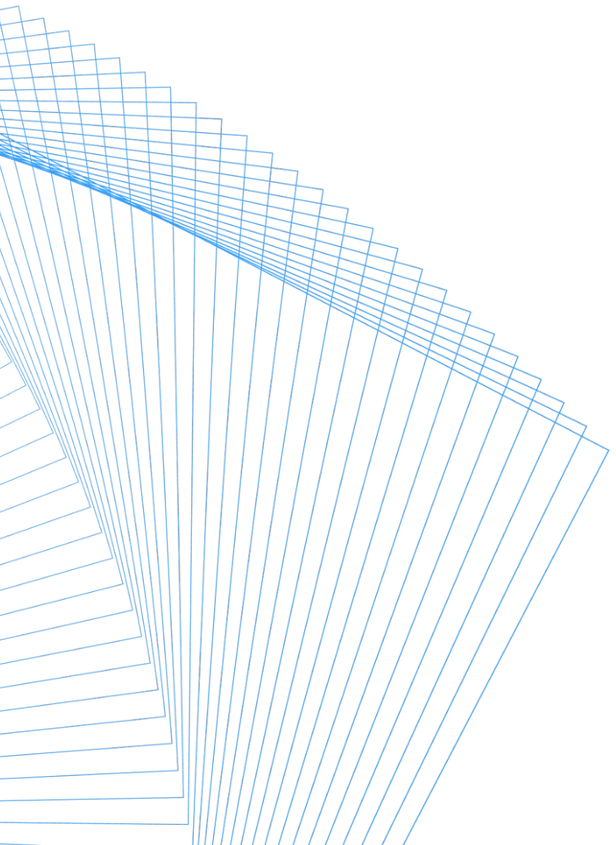
Что делать?

Применить платформенные подходы

Они же — Platform Engineering

Deckhouse
Development Platform

Применить платформенный подход



- Kubernetes — это тоже платформа, но она развивалась параллельно с практиками platform engineering
- Платформа автоматизирует не только развёртывание микросервисов, но и настройку стандартных окружений и инструментов
- Инфраструктурные команды не делают инструменты, а предоставляют услуги (сервисы)
- Повышаем межкомандное взаимодействие за счёт сервисов самообслуживания и enabling teams, применяя модель командных топологий *
- Строим внутреннюю платформу, объединяющую разрозненные инструменты и процессы в единый интерфейс
- Отношение к сервисам инфраструктурных команд как к прозрачному продукту
- Бизнес — внутренний заказчик. Разработчик — пользователь

* Team Topologies

Один из инструментов для реализации платформенного подхода — **IDP**
Internal Developer Platform

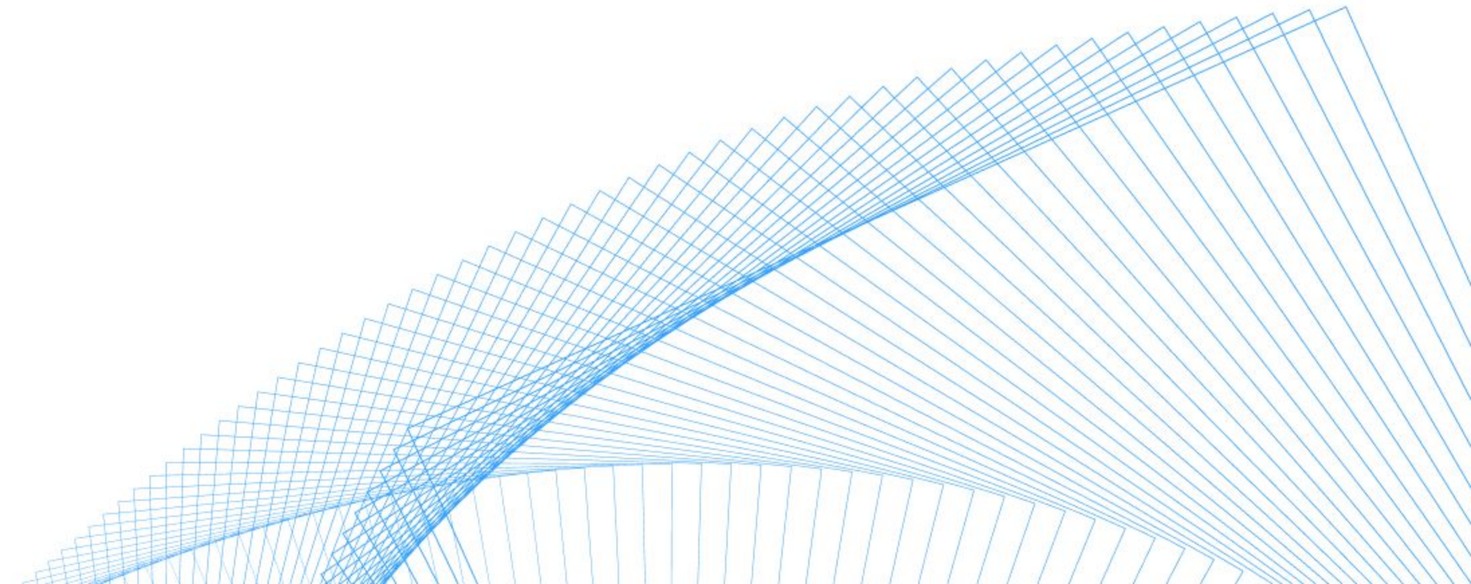
Что такое платформа внутренней разработки (IDP)

Платформа

продукт, дающий один набор согласованных возможностей

Внутренняя платформа разработки (IDP)

оркестратор платформ. Платформа для разработчиков внутри компании



Как цели достигаются при внедрении IDP

Скорость поставки ПО

Снижение времени на проверку продуктовых гипотез

Масштабируемость

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

Снижение стоимости владения

Управление коммунальными ресурсами, мультитенантность, уменьшение дублирования инфраструктуры

Доступность

Предоставление «из коробки» инструментов для повышения доступности приложений

Гибкость

Платформа выступает рычагом для быстрой адаптации команд разработки к новым технологиям

Качество обслуживания

Единое окно для автоматизации и сопровождения процессов разработки

Достигаемые цели

Меньше ручных шагов в релизном цикле

Окружения, конфигурации, доступы и деплой — через самообслуживание и автоматизацию

Быстрее старт новых продуктов и команд

Шаблоны, готовые интеграции и базовая инфраструктура предоставляются платформой как продукт

Меньше ожидания платформенной команды

Стандарты и типовые практики доступны командам разработки «по кнопке», без очередей и ручной поддержки

Меньше возвратов на доработку

Встроенные проверки, политики и стандартные пайплайны снижают количество ошибок и повторной работы

Достигаемые цели: до 224 % ROI *

Меньше инструментов | Меньше ручных операций | Больше предсказуемости

Снижает операционные усилия на приложение

Самообслуживание и стандартизация уменьшают ручные операции DevOps/SRE

Снижает стоимость изменений через «эталонные пути»

Типовые решения вместо уникальных сборок в каждой команде — меньше переделок и быстрее поставка

Снижает риск и длительность инцидентов

Стандарты и наблюдаемость «по умолчанию» ускоряют поиск контекста и владельцев, упрощают обновления

Повышает предсказуемость затрат

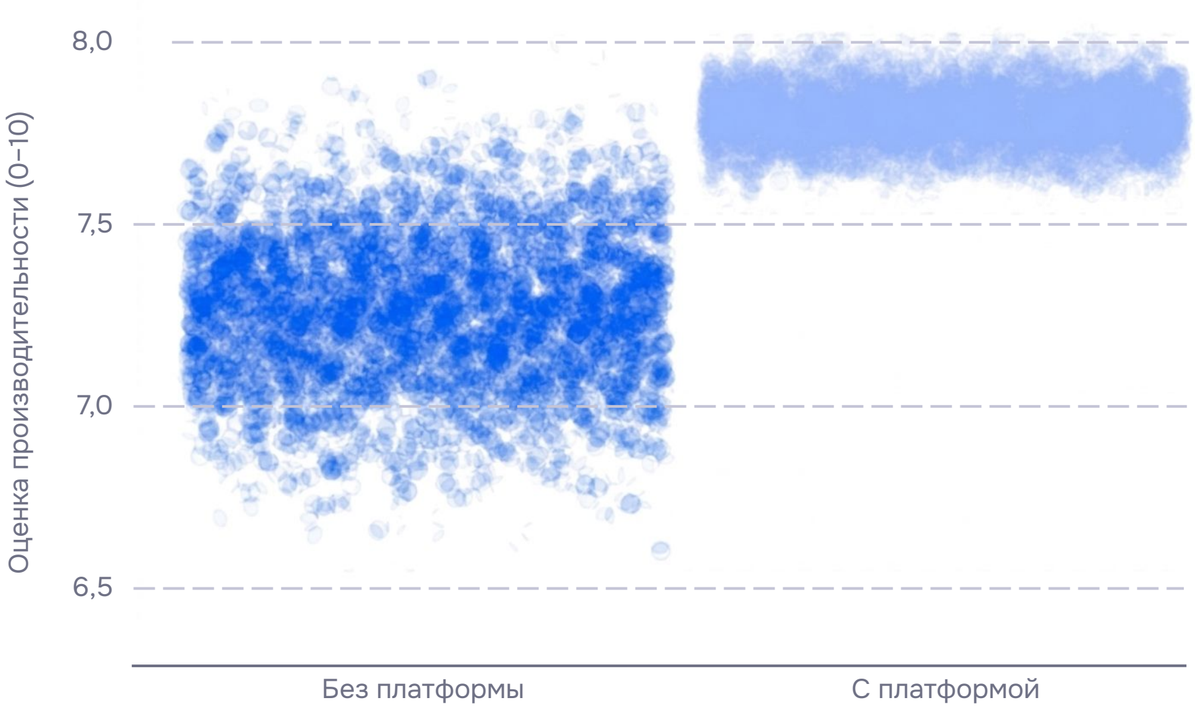
Единая платформа проще планируется и контролируется, чем набор разрозненных инструментов

* Forrester TEI

Внутренние платформы разработки

выбор лидеров индустрии и ключ
к технологическому превосходству

DORA Accelerate State of DevOps 2024



Ключевые преимущества платформенной инженерии для разработчиков

State of DevOps Russia 2025

50 %
Повышение продуктивности

40 %
Более высокое качество программного обеспечения

36 %
Сокращение времени развёртывания

36 %
Более стабильные приложения

31 %
Меньше ошибок

31 %
Снижение затрат

31 %
Сокращение времени разработки продукта

31 %
Снижение риска утечек безопасности

29%
Возможность для разработчиков выполнять несколько задач

29 %
Повышение удовлетворённости клиентов

28 %
Сокращение времени на обслуживание ПО

27 %
Снижение когнитивной нагрузки

24 %
Сокращение времени, затрачиваемого на безопасность

22 %
Улучшенная переносимость приложения

Как обеспечить внедрение качественной платформы (IDP)

Deckhouse
Development Platform

Атрибуты качественной платформы: CNCF

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

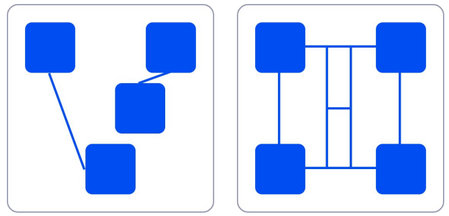
Опциональная и комбинированная

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

Безопасная по умолчанию

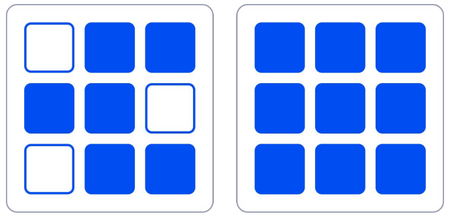
Должна быть безопасной «из коробки» и позволять
проверять систему на соответствие политикам
и стандартам, принятым в организации

Атрибуты качественной платформы: G. Hohpe



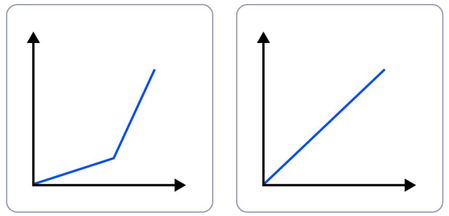
Связность

Представляет ли платформа собой целостное и значимое целое, в отличие от разрозненной коллекции компонентов?



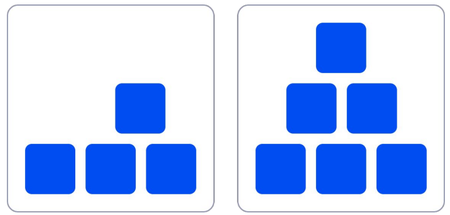
Синергия

Содержит ли платформа полный набор сервисов или только частично покрывает технологический ландшафт?



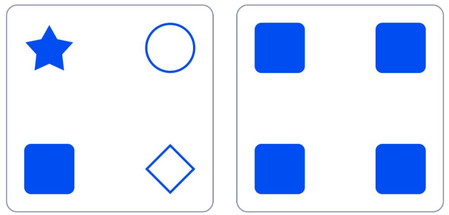
Соразмерная ценность

Получают ли пользователи пропорциональную ценность от использования лишь части платформы?



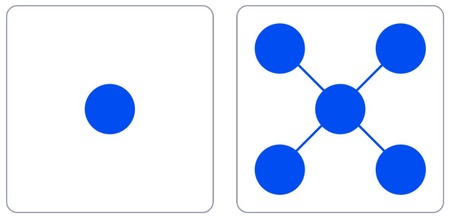
Самодостаточность

Обеспечивает ли платформа полный спектр возможностей?



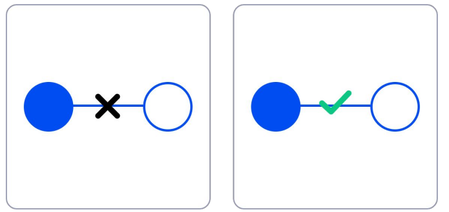
Согласованность

Может ли пользователь, изучив одну часть платформы, применять приобретённые знания к другим её частям?



Связанность

Насколько платформа зависима от внешних систем?



Свободный переход

Насколько легко отказаться от использования платформы, если это потребуется?

Как внедрить IDP

Deckhouse
Development Platform

Варианты внедрения IDP

Build

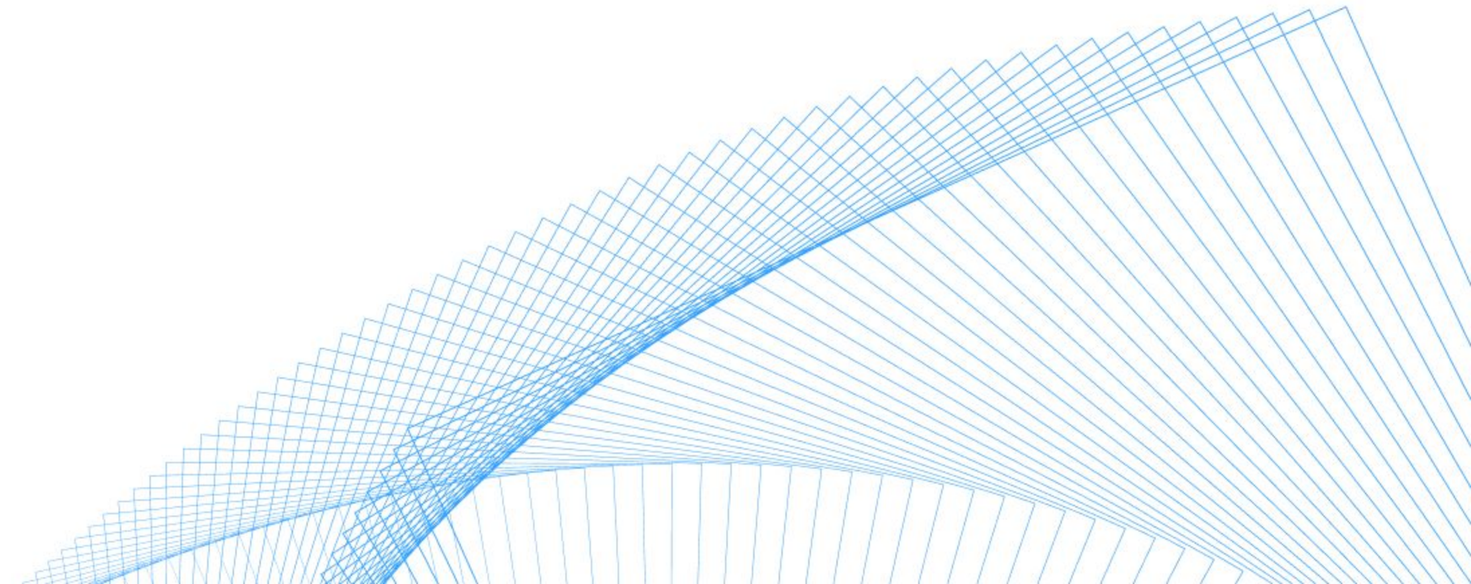
Portal + Backend

Customize

OSS | Vendor

Buy

On-Prem | PaaS



Подводные камни при построении IDP «с нуля»

Время на внедрение

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

Масштабируемость

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

Стоимость

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

Риски поддержки

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

Гибкость

Высокая. Нет технических ограничений

Контроль

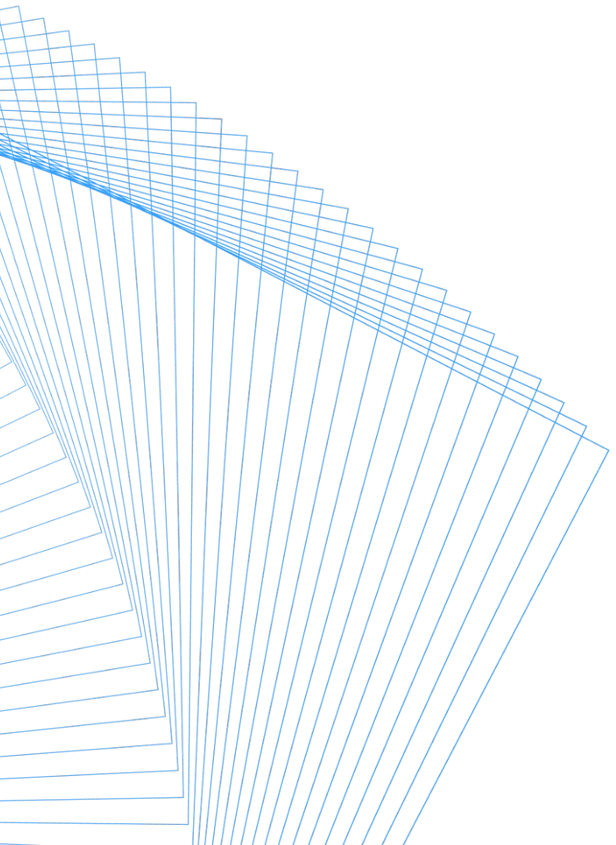
Полный контроль над архитектурой решения, безопасностью, данными, зависимостями

Обзор продукта Deckhouse Development Platform (DDP)

Deckhouse
Development Platform

Deckhouse Development Platform

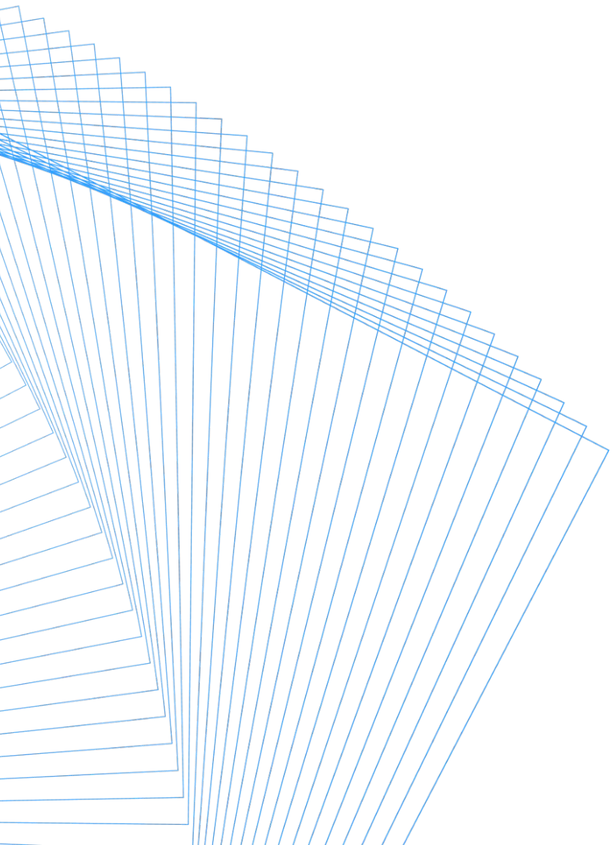
IDP, которая ускоряет и упрощает процессы разработки



- Служит единым центром доступа ко всем необходимым инструментам и ресурсам
- Снижает нагрузку на команды разработчиков и DevOps
- Унифицирует и автоматизирует процессы, обеспечивая прозрачность и контроль над всеми аспектами разработки и развёртывания
- За счёт единой документации и инструментов улучшает процесс онбординга новых сотрудников
- Предоставляет актуальную информацию о приложениях, связях, статусе и т. д.

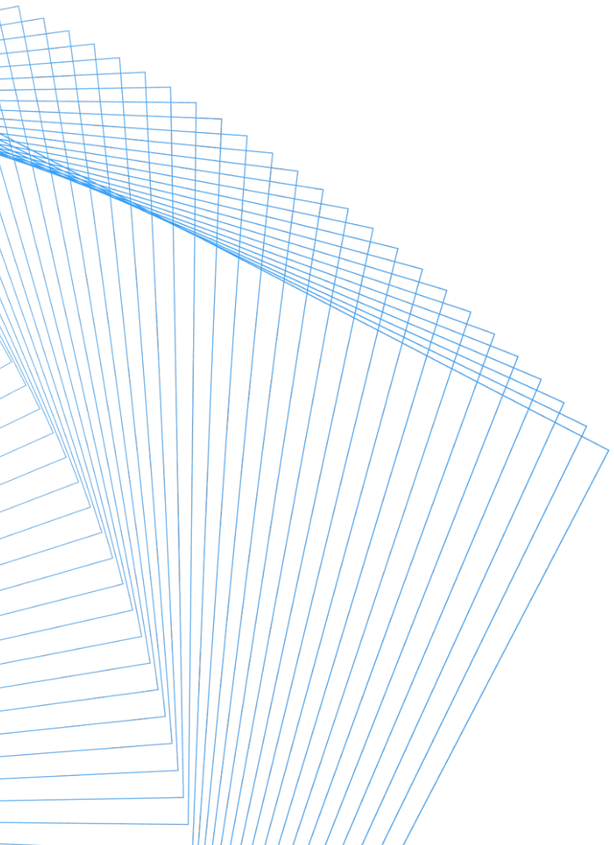
Deckhouse Development Platform

Комплексное решение
для управления всеми этапами
разработки через единое окно



- Благодаря гибкости и расширяемости позволяет оцифровать любые сценарии разработки
- Максимальные возможности кастомизации портала
- Встраивается в любой существующий стек и интегрируется с любыми системами через REST API
- Сразу создавалась как продукт по принципам, принятым в мировой индустрии, нет привязки к процессам конкретной компании

Основной функционал



01 Портал разработки

Сервисный каталог, связи, дашборды и виджеты

02 Оркестратор разработки

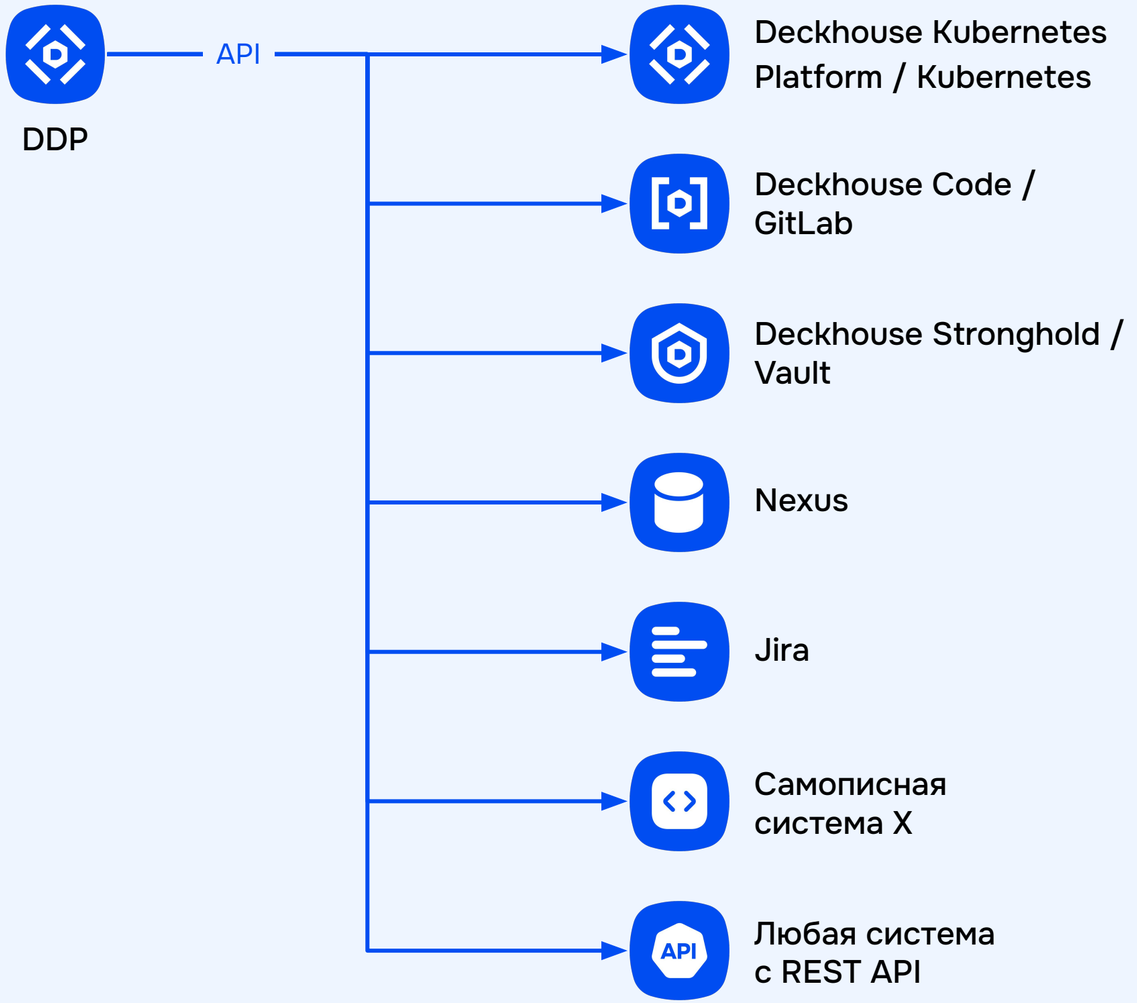
Платформа автоматизации (в том числе сценариев разработки), интеграционная панель, self-service

03 AI-ассистент и MCP

Доступ к AI-ассистенту из веб-интерфейса платформы, MCP-прокси для внешних сервисов

Как это работает

- Опрашивает системы через API, формирует сервисный каталог и строит связи
- Выводит дашборды с основной информацией из систем
- Проверяет статус ресурсов в системах
- Предоставляет self-service-механизмы для изменения в системах
- Предлагает MCP-прокси для получения данных из систем



Помогает ИТ-специалистам работать эффективнее

Автоматизирует операции, связанные с жизненным циклом продуктов, и предоставляет инструменты для оценки работы всей системы

Единые стандарты разработки и эталонный путь

Платформа позволяет подготовить шаблоны и типовые конфигурации (CI/CD, Kubernetes и т. д.) и обеспечивает трансляцию стандартов во все команды разработки.

Улучшает:

- Lead Time
- Time to First Value

Интеграция неинтегрируемых сервисов

Взаимодействие можно настроить с любой инфраструктурной системой, где есть API (система управления заявками, открытие доступов в файрвол и т. д.).

Улучшает:

- Cross-Team Handoff Time
- Change Failure Rate

Упрощённый онбординг разработчиков и команд

Новые сотрудники могут быстро начать работу за счёт шаблонных решений, документации и преднастроенной структуры проектов, которые предоставляются через платформу.

Улучшает:

- Time to Onboard
- Time to First Value

Контроль качества

Решение предоставляет механизм Quality Gates и быстро верифицирует ключевые параметры работы подключённых сервисов.

Улучшает:

- MTTR
- Change Failure Rate

Проблематика

Deckhouse
Development Platform

Менеджмент

Недостаток проникновения стандартов и согласованности

«Разные команды используют разные технологии и процессы, что снижает эффективность».

Вводит стандарты и унифицированные процессы, обеспечивая согласованность и совместимость

Проблемы с видимостью и контролем

«Сложно отслеживать состояние разработки, развёртывания, производительности приложений».

Предоставляет централизованные дашборды и отчёты, обеспечивая прозрачность и полноценный контроль над процессами

Сохранение темпов разработки и качества продукта в условиях сокращения затрачиваемых ресурсов

«Как обеспечить ту же скорость поставки и уровень качества меньшими ресурсами?»

Повышает производительность команды за счёт стандартизации, self-service и автоматизации, позволяя меньшими ресурсами поддерживать тот же темп разработки и уровень качества

Проблематика

Deckhouse
Development Platform

Команды разработки

Долгий процесс коммуникации и согласования, трудности с доступом к инструментам

«Я хочу получить БД и жду два месяца».

Предоставляет единый интерфейс для доступа ко всем необходимым инструментам и информации о них, упрощая их использование

Сложные и неочевидные процессы разработки / сложность настройки окружения

«Что мне надо сделать, чтобы создать и развернуть сервис?»

Автоматизирует настройку окружения, предоставляя стандартные конфигурации и инструменты, что ускоряет начало работы и минимизирует ошибки

Непрозрачность связей между ресурсами и зависимостями

«Какой Docker registry использует сервис?» «Где развёрнут сервис, а где нет?»

Предоставляет автоматизированное создание и учёт связей между различными ресурсами инфраструктуры

Проблематика

Deckhouse
Development Platform

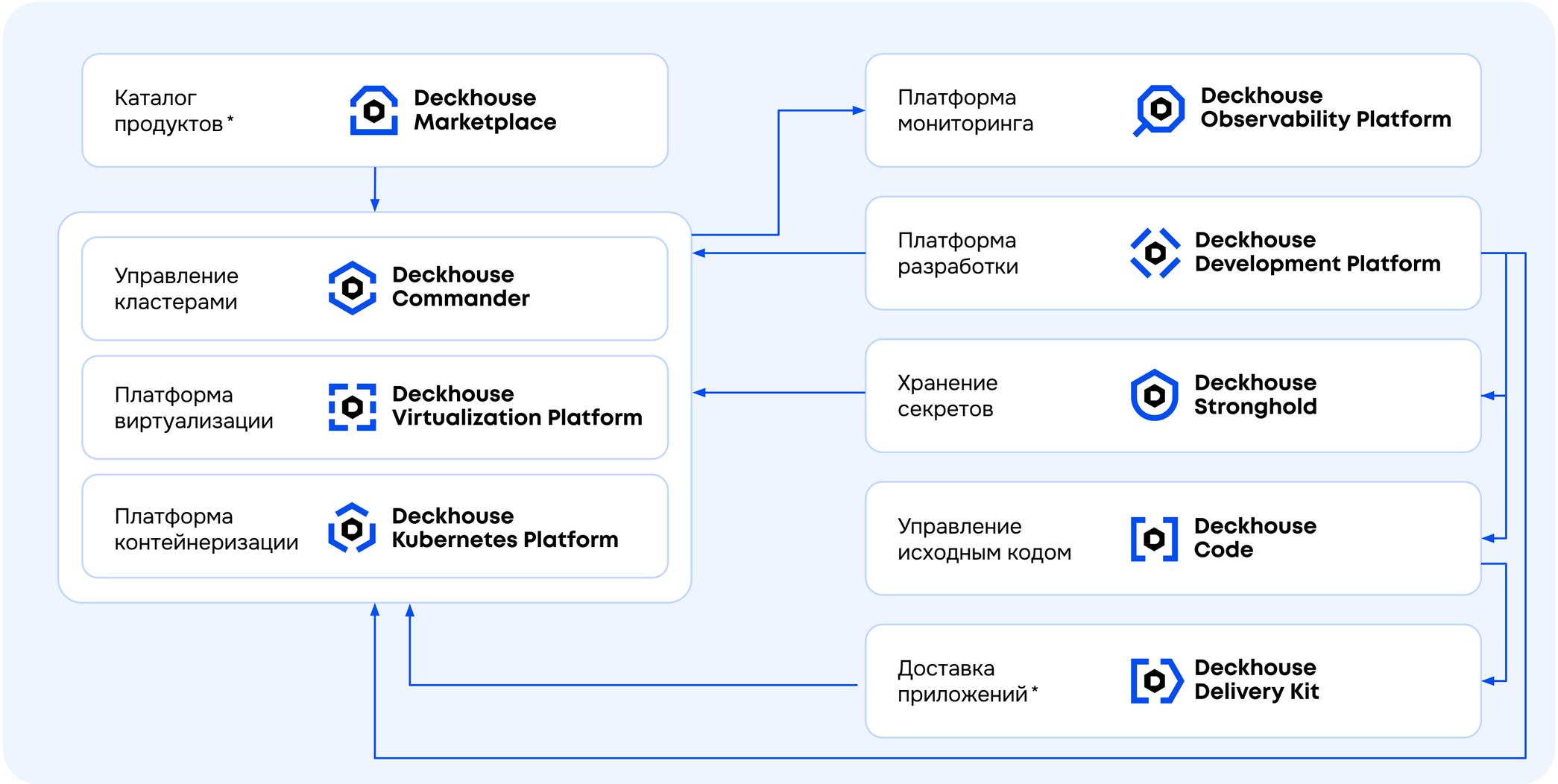
Инфраструктурные команды

Высокая нагрузка на инфраструктурные команды

«Текущее количество задач и объём коммуникаций, связанных с поддержкой инфраструктуры, тормозят работу инфраструктурных команд».

Автоматизирует рутинные задачи, освобождая ресурсы инфраструктурных команд для более стратегических инициатив

Экосистема продуктов Deckhouse



* Продукт находится в активной фазе разработки, готовится выход релизной версии

Возможные траектории развития DDP:

01 SDLC-мониторинг

- Автоматизированный сбор и анализ событий SDLC из Git, таск-трекера, пайплайнов и инфраструктуры
- Подсчёт DORA-метрик, показателей зрелости и качества
- Подсчёт стоимости инфраструктуры и человеческих ресурсов (TCO)
- Автоматическая идентификация технического долга и отслеживание прогресса устранения
- Дашборды и отчёты в разрезах структуры организации и команд

02 Развитие интеграции с AI-инструментами

- Troubleshooting-ассистент
- Onboarding-ассистент для недавно нанятых разработчиков
- Поисковик и саммаризатор по дашбордам, сервис-каталогу, таск-трекеру, пайплайнам, Git и т. п.

03 Разработка безопасного программного обеспечения

- Наблюдаемость процессов безопасной разработки
- Автоматическая идентификация частей ландшафта, не удовлетворяющих требованиям ИБ
- Понятные чек-листы и планы работ к исполнению требований регулятора (РБПО)

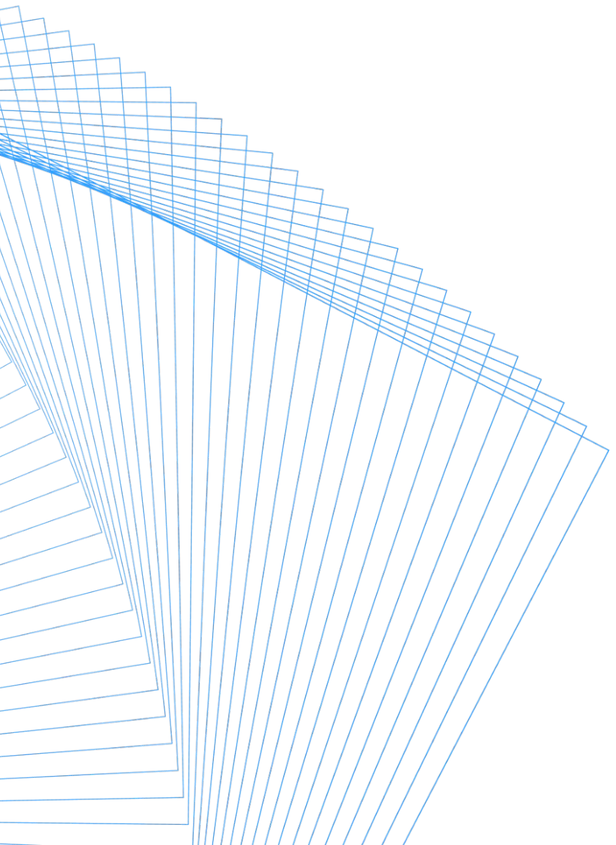
04 PaaS для прототипов и небольших проектов

- Облегчённый инструментарий для разработчиков-неэкспертов
- Поддержка «из коробки» популярных фреймворков и нетиповых приложений
- Ограниченный, но быстрый и безопасный процесс, деплой за минуты

Выводы

Deckhouse
Development Platform

Зачем вам IDP?



01 Повысить профиль эффективности

- Усилить инфраструктурный ландшафт

02 Развить платформенную инженерию

- Продуктовый подход к инфраструктурным сервисам
- Клиентоцентричность
- Усилить внутренние взаимодействия (Team Topologies)

03 Закрыть потребности менеджмента

- Повысить видимость и контроль
- Стандартизировать процессы и инструменты
- Повысить производительность команд

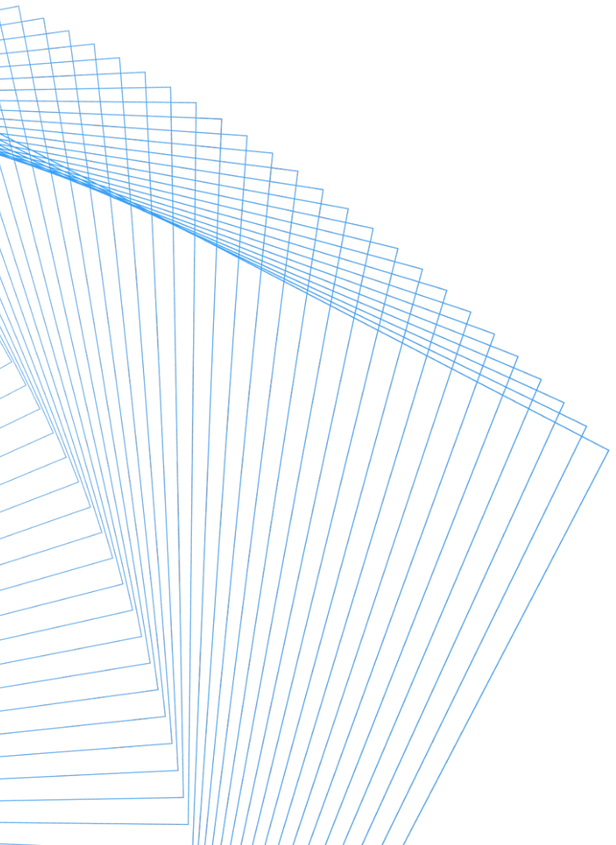
04 Закрыть потребности разработки

- Единый интерфейс взаимодействия
- Автоматизация работы с окружениями
- Ускорение онбординга

05 Закрыть потребности инфраструктуры

- Снижение рутинной нагрузки

Следующие шаги



01 ## Важно

- Сохранять атрибуты качества платформы
- Развивать практики Platform Engineering
- Соблюдать качественные подходы по взаимодействию команд с применением Team Topologies

02 ## Диагностика IDP

- Как развивать IDP в условиях вашей компании?
- Как улучшить метрики эффективности разработки?
- Как интегрировать неинтегрируемые процессы и инструменты разработки?

03 ## Внедрение Deckhouse Development Platform

- Бесплатная консультация с нашим экспертом (первые 10 заявок)
- Специальные условия на услуги внедрения (до 31 мая 2026 года)

Deckhouse
Development Platform

Спасибо
за внимание!

Deckhouse Development Platform: описание продукта

Оставить заявку на консультацию

contact@deckhouse.ru

+7 (495) 721-10-27

deckhouse.ru