

Микросервисное безумие и что из нас делает solution'ов

Чернухин Максим
СТО Клиентского Сервиса
Сбер Страхование Жизни

Чернухин Максим

- 10 лет в
 - 3 линия поддержки
 - Back, Front, ML и тд
 - Архитектура, инфраструктура
 - СТО
- Люблю ТРИЗ и сложные задачи
- Нравится восходить на горы
- Есть 30 и 40-летний бонсай



facebook.com/chernukhin.maksim

TG @MaksCher

О чём сегодня поговорим

- Что такое системный анализ в разработке
- Интеграция с внешними системами компании
- В чём разница Solution архитектора и Аналитика
- Что поменялось с приходом микросервисами
- Создание среды помогающей решать задачи
- Итоги

О чём сегодня поговорим

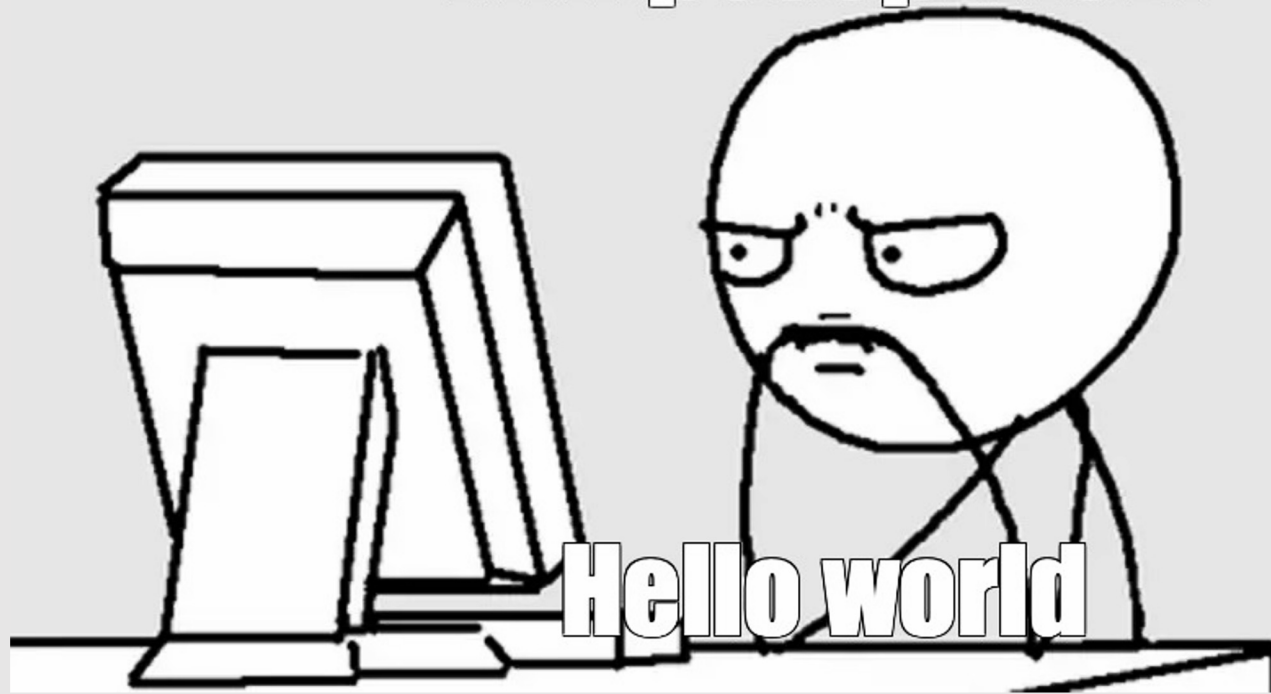
- Что такое системный анализ в разработке
- Интеграция с внешними системами компании
- В чём разница Solution архитектора и Аналитика
- Что поменялось с приходом микросервисами
- Создание среды помогающей решать задачи
- Итоги

Небольшая команда, стартуем разработку

- Команда до 10 человек
- Пару микросервисов, а может и вовсе нет

Старт разработки

Микросервисы



Провести системный анализ

- Сбор и анализ требований от заказчика

Провести системный анализ

- Сбор и анализ требований от заказчика
- Согласование требований

Провести системный анализ

- Сбор и анализ требований от заказчика
- Согласование требований
- Разработка требований к ПО

Провести системный анализ

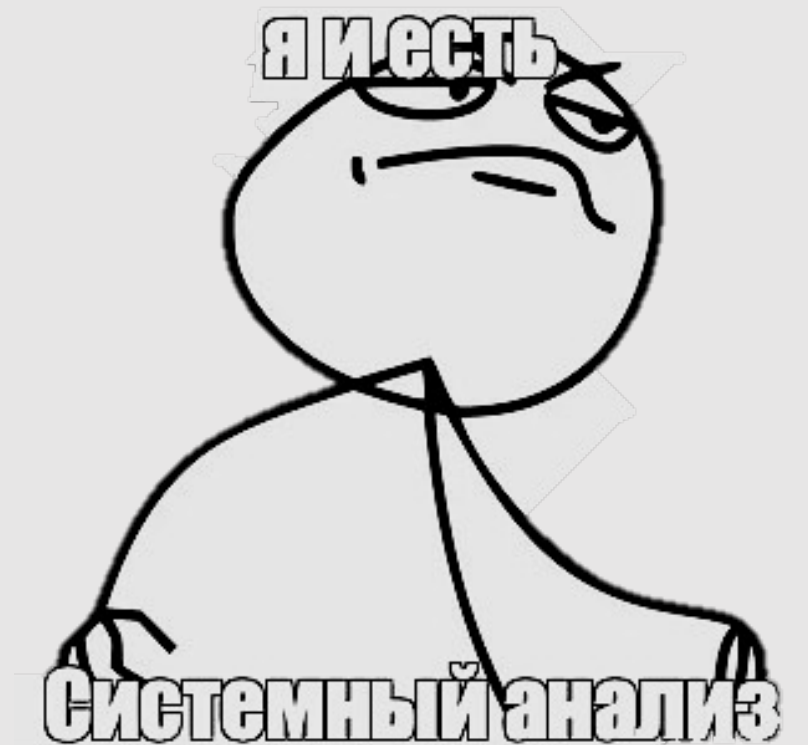
- Сбор и анализ требований от заказчика
- Согласование требований
- Разработка требований к ПО
 - Описание API
 - Диаграмма последовательностей
 - ER диаграмма БД

Провести системный анализ

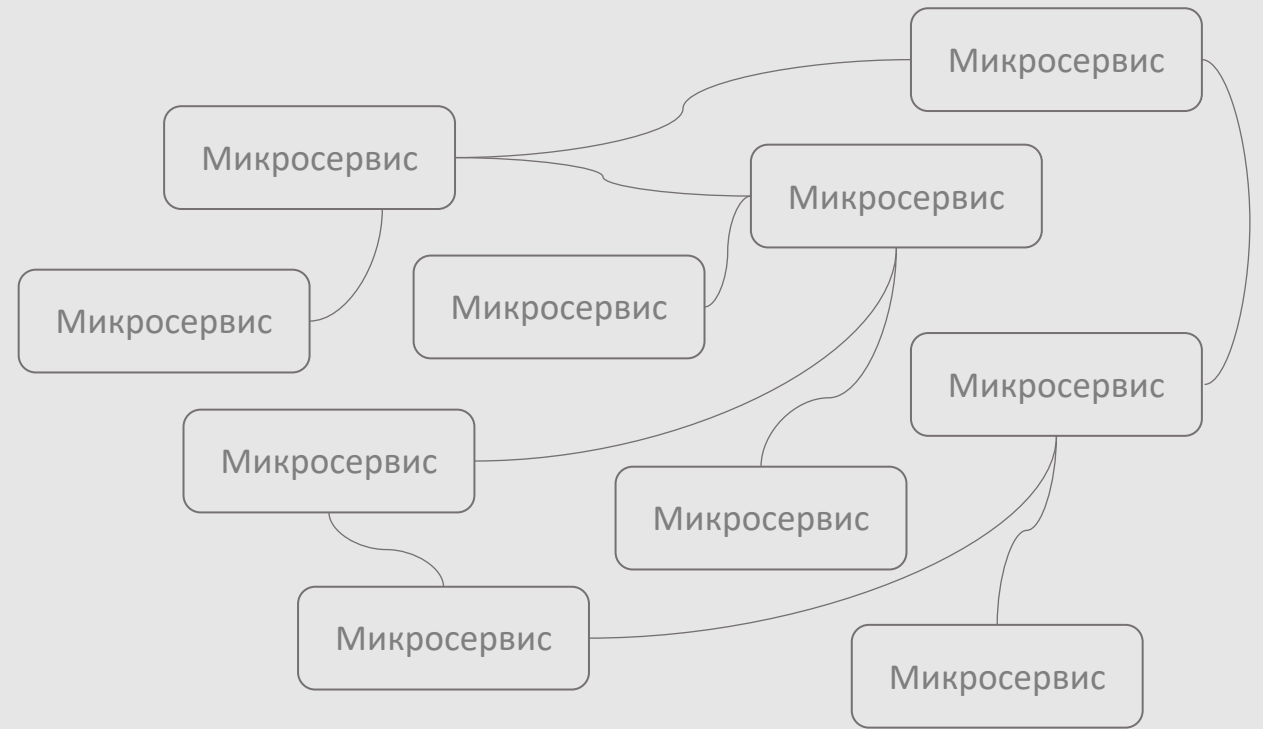
- Сбор и анализ требований от заказчика
- Согласование требований
- Разработка требований к ПО
 - Описание API
 - Диаграмма последовательностей
 - ER диаграмма БД
- Начать разработку

Провести системный анализ

- Сбор и анализ требований от заказчика
- Согласование требований
- Разработка требований к ПО
 - Описание API
 - Диаграмма последовательностей
 - ER диаграмма БД
- Начать разработку



30 сервисов



Провести системный анализ

- Сбор и анализ требований от заказчика
- Согласование требований
- Разработка требований к ПО
 - Описание API
 - Диаграмма последовательностей
 - ER диаграмма БД
- Отдать в разработку

Провести системный анализ

- Сбор и анализ требований от заказчика
- Согласование требований
- **Анализ ландшафта и понимание того, что надо доработать**
- Разработка требований к ПО
 - Описание API
 - Диаграмма последовательностей
 - ER диаграмма БД
- Отдать в разработку

Провести системный анализ

- Сбор и анализ требований от заказчика
- Согласование требований
- **Анализ ландшафта и понимание того, что надо доработать**
- **Разработка требований к ПО с общими правилами**
 - **Описание API и общая обработка ошибок, формат контракта и тд**
 - Диаграмма последовательностей
 - ER диаграмма БД
- Отдать в разработку

Что ещё надо не забыть

- Зоны ответственности сервисов

Что ещё надо не забыть

- Зоны ответственности сервисов
- Принятые шаблоны проектирования

Что ещё надо не забыть

- Зоны ответственности сервисов
- Принятые шаблоны проектирования
- Как разрабатываем BPM сервисы

Что ещё надо не забыть

- Зоны ответственности сервисов
- Принятые шаблоны проектирования
- Как разрабатываем BPM сервисы
- Где нужен оркестратор

Что ещё надо не забыть

- Зоны ответственности сервисов
- Принятые шаблоны проектирования
- Как разрабатываем BPM сервисы
- Где нужен оркестратор
- Общий формат ошибок

Что ещё надо не забыть

- Зоны ответственности сервисов
- Принятые шаблоны проектирования
- Как разрабатываем BPM сервисы
- Где нужен оркестратор
- Общий формат ошибок
- Как настроить фильтрацию для REST запросов

Что ещё надо не забыть

- Зоны ответственности сервисов
- Принятые шаблоны проектирования
- Как разрабатываем BPM сервисы
- Где нужен оркестратор
- Общий формат ошибок
- Как настроить фильтрацию для REST запросов
- Как выбрать синхрон или асинхрон

Что ещё надо не забыть

- Зоны ответственности сервисов
- Принятые шаблоны проектирования
- Как разрабатываем BPM сервисы
- Где нужен оркестратор
- Общий формат ошибок
- Как настроить фильтрацию для REST запросов
- Как выбрать синхрон или асинхрон
- Не забыть про логировани и аудит

Что ещё надо не забыть

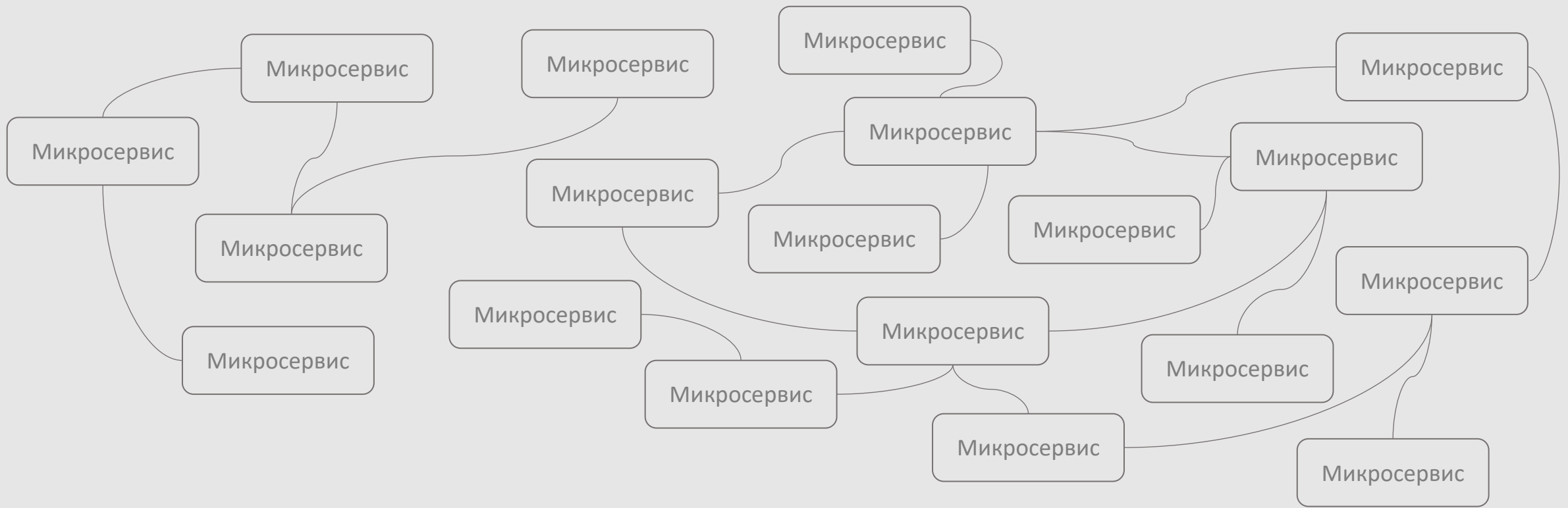
- Зоны ответственности сервисов
- Принятые шаблоны проектирования
- Как разрабатываем BPM сервисы
- Где нужен оркестратор
- Общий формат ошибок
- Как настроить фильтрацию для REST запросов
- Как выбрать синхрон или асинхрон
- Не забыть про логировани и аудит
- Хранение файлов, проверка на вирусы

Что ещё надо не забыть

- Зоны ответственности сервисов
- Принятые шаблоны проектирования
- Как разрабатываем BPM сервисы
- Где нужен оркестратор
- Общий формат ошибок
- Как настроить фильтрацию для REST запросов
- Как выбрать синхрон или асинхрон
- Не забыть про логировани и аудит
- Хранение файлов, проверка на вирусы
- Аутентификация и авторизация

О чём сегодня поговорим

- Что такое системный анализ в разработке
- **Интеграция с внешними системами компании**
- В чём разница Solution архитектора и Аналитика
- Что поменялось с приходом микросервисами
- Создание среды помогающей решать задачи
- Итоги

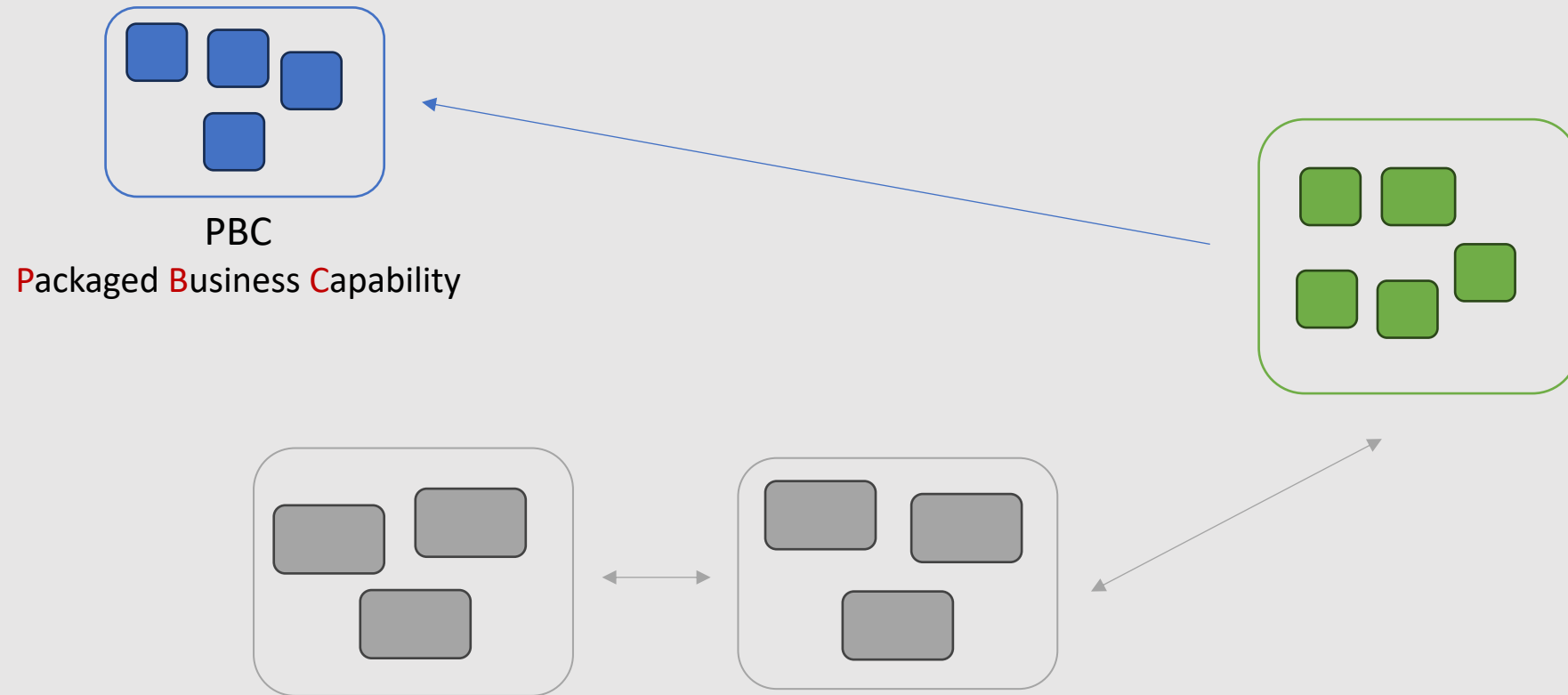


100+ сервисов

Большая команда, развитие продукта

- Команд 10+
- Необходимо сделать интеграцию с внешней системой

Внешняя интеграция с другим бизнес кластером



Внешняя интеграция с другим бизнес кластером

- Необходимо найти нужную нам систему

Внешняя интеграция с другим бизнес кластером

- Необходимо найти нужную нам систему
- Нам надо узнать правила подключения, допустимую нагрузку

Внешняя интеграция с другим бизнес кластером

- Необходимо найти нужную нам систему
- Нам надо узнать правила подключения, допустимую нагрузку
- Нам надо узнать правила логирования и аудита

Внешняя интеграция с другим бизнес кластером

- Необходимо найти нужную нам систему
- Нам надо узнать правила подключения, допустимую нагрузку
- Нам надо узнать правила логирования и аудита
- Получить доступ: аутентификация и авторизация

Внешняя интеграция с другим бизнес кластером

- Необходимо найти нужную нам систему
- Нам надо узнать правила подключения, допустимую нагрузку
- Нам надо узнать правила логирования и аудита
- Получить доступ: аутентификация и авторизация
- Общий подход к ошибкам

Внешняя интеграция с другим бизнес кластером

- Необходимо найти нужную нам систему
- Нам надо узнать правила подключения, допустимую нагрузку
- Нам надо узнать правила логирования и аудита
- Получить доступ: аутентификация и авторизация
- Общий подход к ошибкам
- Единый подход к контрактам

Кто может ответить на этот вопрос?



Где мастер-система!?

Кто может ответить на этот вопрос?

- Solution архитектор
- И команда, которая за неё отвечает

О чём сегодня поговорим

- Что такое системный анализ в разработке
- Интеграция с внешними системами компании
- В чём разница Solution архитектора и Аналитика
- Что поменялось с приходом микросервисами
- Создание среды помогающей решать задачи
- Итоги

А теперь давайте сравним

- Необходимо найти нужную нам систему
- Нам надо узнать правила подключения, допустимую нагрузку
- Нам надо узнать правила логирования и аудита
- Получить доступ: аутентификация и авторизация
- Общий подход к ошибкам
- Единый подход к контрактам
- Зоны ответственности сервисов
- Как выбрать синхрон или асинхрон
- Хранение файлов, проверка на вирусы
- Не забыть про логировани и аудит
- Аутентификация и авторизация
- Общий формат ошибок
- Как настроить фильтрацию для REST запросов
- Принятые шаблоны проектирования
- Как разрабатываем BPM сервисы
- Где нужен оркестратор

А теперь давайте сравним

- Необходимо найти нужную нам систему
- Нам надо узнать правила подключения, допустимую нагрузку
- Нам надо узнать правила логирования и аудита
- Получить доступ: аутентификация и авторизация
- Общий подход к ошибкам
- Единый подход к контрактам
- Зоны ответственности сервисов
- Как выбрать синхрон или асинхрон
- Хранение файлов, проверка на вирусы
- Не забыть про логировани и аудит
- Аутентификация и авторизация
- Общий формат ошибок
- Как настроить фильтрацию для REST запросов
- Принятые шаблоны проектирования
- Как разрабатываем BPM сервисы
- Где нужен оркестратор

А теперь давайте сравним

- Необходимо найти нужную нам систему
- Нам надо узнать правила подключения, допустимую нагрузку
- **Нам надо узнать правила логирования и аудита**
- Получить доступ: аутентификация и авторизация
- Общий подход к ошибкам
- Единый подход к контрактам
- Зоны ответственности сервисов
- Как выбрать синхрон или асинхрон
- Хранение файлов, проверка на вирусы
- **Не забыть про логировани и аудит**
- Аутентификация и авторизация
- Общий формат ошибок
- Как настроить фильтрацию для REST запросов
- Принятые шаблоны проектирования
- Как разрабатываем BPM сервисы
- Где нужен оркестратор

А теперь давайте сравним

- Необходимо найти нужную нам систему
- Нам надо узнать правила подключения, допустимую нагрузку
- Нам надо узнать правила логирования и аудита
- **Получить доступ: аутентификация и авторизация**
- Общий подход к ошибкам
- Единый подход к контрактам
- Зоны ответственности сервисов
- Как выбрать синхрон или асинхрон
- Хранение файлов, проверка на вирусы
- Не забыть про логировани и аудит
- **Аутентификация и авторизация**
- Общий формат ошибок
- Как настроить фильтрацию для REST запросов
- Принятые шаблоны проектирования
- Как разрабатываем BPM сервисы
- Где нужен оркестратор

А теперь давайте сравним

- Необходимо найти нужную нам систему
- Нам надо узнать правила подключения, допустимую нагрузку
- Нам надо узнать правила логирования и аудита
- Получить доступ: аутентификация и авторизация
- **Общий подход к ошибкам**
- Единый подход к контрактам
- Зоны ответственности сервисов
- Как выбрать синхрон или асинхрон
- Хранение файлов, проверка на вирусы
- Не забыть про логировани и аудит
- Аутентификация и авторизация
- **Общий формат ошибок**
- Как настроить фильтрацию для REST запросов
- Принятые шаблоны проектирования
- Как разрабатываем BPM сервисы
- Где нужен оркестратор

А теперь давайте сравним

- Необходимо найти нужную нам систему
- Нам надо узнать правила подключения, допустимую нагрузку
- Нам надо узнать правила логирования и аудита
- Получить доступ: аутентификация и авторизация
- Общий подход к ошибкам
- **Единый подход к контрактам**
- Зоны ответственности сервисов
- Как выбрать синхрон или асинхрон
- Хранение файлов, проверка на вирусы
- Не забыть про логировани и аудит
- Аутентификация и авторизация
- Общий формат ошибок
- **Как настроить фильтрацию для REST запросов**
- Принятые шаблоны проектирования
- Как разрабатываем BPM сервисы
- Где нужен оркестратор

А теперь давайте сравним

- Необходимо найти нужную нам систему
- Нам надо узнать правила подключения, допустимую нагрузку
- Нам надо узнать правила логирования и аудита
- Получить доступ: аутентификация и авторизация
- Общий подход к ошибкам
- Единый подход к контрактам
- ????
- Зоны ответственности сервисов
- Как выбрать синхрон или асинхрон
- Хранение файлов, проверка на вирусы
- Не забыть про логировани и аудит
- Аутентификация и авторизация
- Общий формат ошибок
- Как настроить фильтрацию для REST запросов
- Принятые шаблоны проектирования
- Как разрабатываем BPM сервисы
- Где нужен оркестратор

А теперь давайте сравним

- Необходимо найти нужную нам систему
- Нам надо узнать правила подключения, допустимую нагрузку
- Нам надо узнать правила логирования и аудита
- Получить доступ: аутентификация и авторизация
- Общий подход к ошибкам
- Единый подход к контрактам
- Создание новой системы
- Зоны ответственности сервисов
- Как выбрать синхрон или асинхрон
- Хранение файлов, проверка на вирусы
- Не забыть про логировани и аудит
- Аутентификация и авторизация
- Общий формат ошибок
- Как настроить фильтрацию для REST запросов
- Принятые шаблоны проектирования
- Как разрабатываем BPM сервисы
- Где нужен оркестратор

И в чем же разница?

Аналитик

- Надо понимать бизнес задачи и понимать как они ложатся на архитектуру
(команды, РВС, бизнес-кластера)

Solution архитектор

- Надо понимать бизнес задачи и понимать как они ложатся на архитектуру **КОМПАНИИ**

И в чем же разница?

Аналитик

- Надо понимать бизнес задачи и понимать как они ложатся на архитектуру (команды, РВС, бизнес-кластера)
- К аналитикам приходят с доработками **бизнес-кластера**, но они могут инициировать общие доработки компании через solution архитектора

Solution архитектор

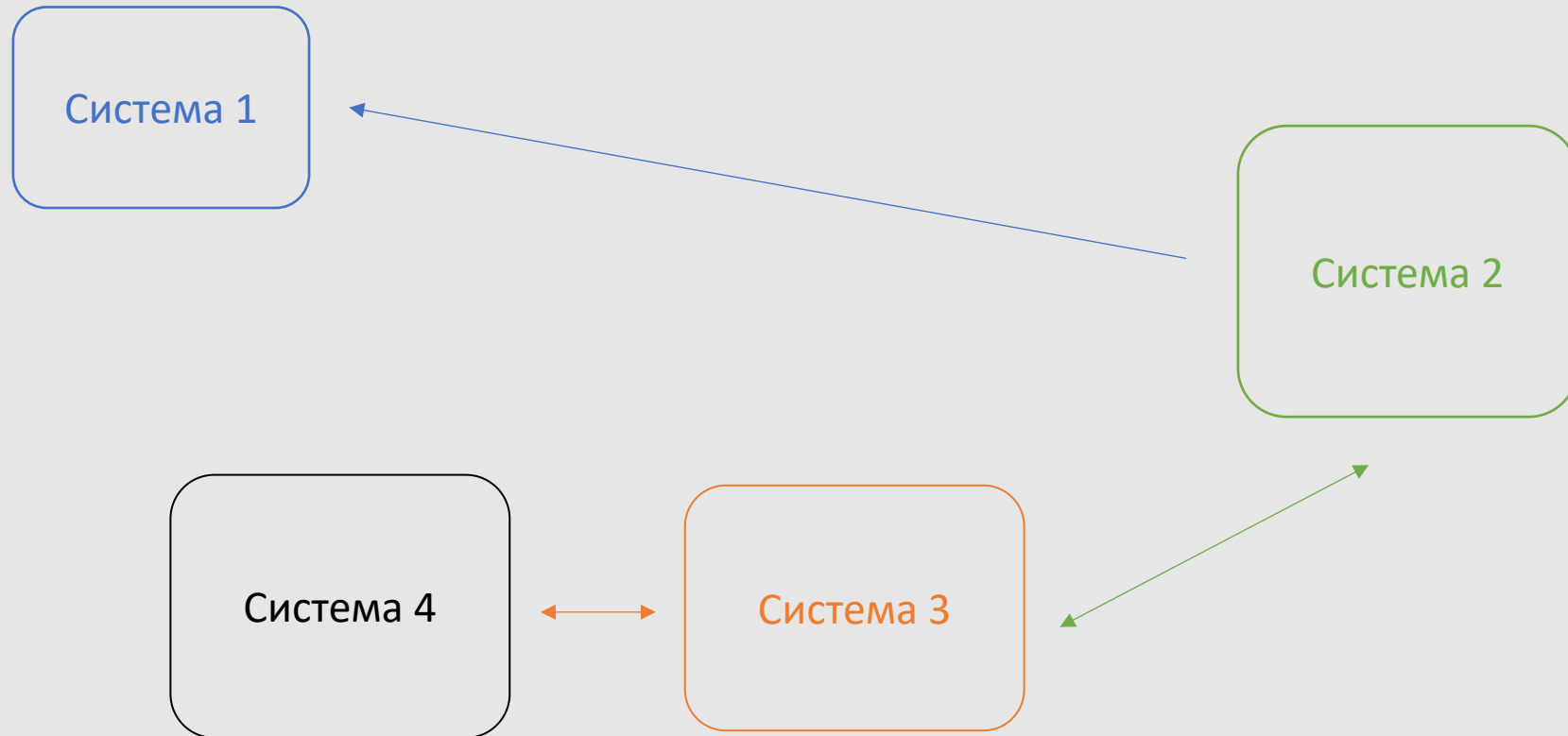
- Надо понимать бизнес задачи и понимать как они ложатся на архитектуру **компании**
- К solution архитекторам приходят, для больших доработок уровня **компании** или **нескольких систем** и они могут отправить задачи на множество команд к аналитикам

О чём сегодня поговорим

- Что такое системный анализ в разработке
- Интеграция с внешними системами компании
- В чём разница Solution архитектора и Аналитика
- **Что поменялось с приходом микросервисами**
- Создание среды помогающей решать задачи
- Итоги

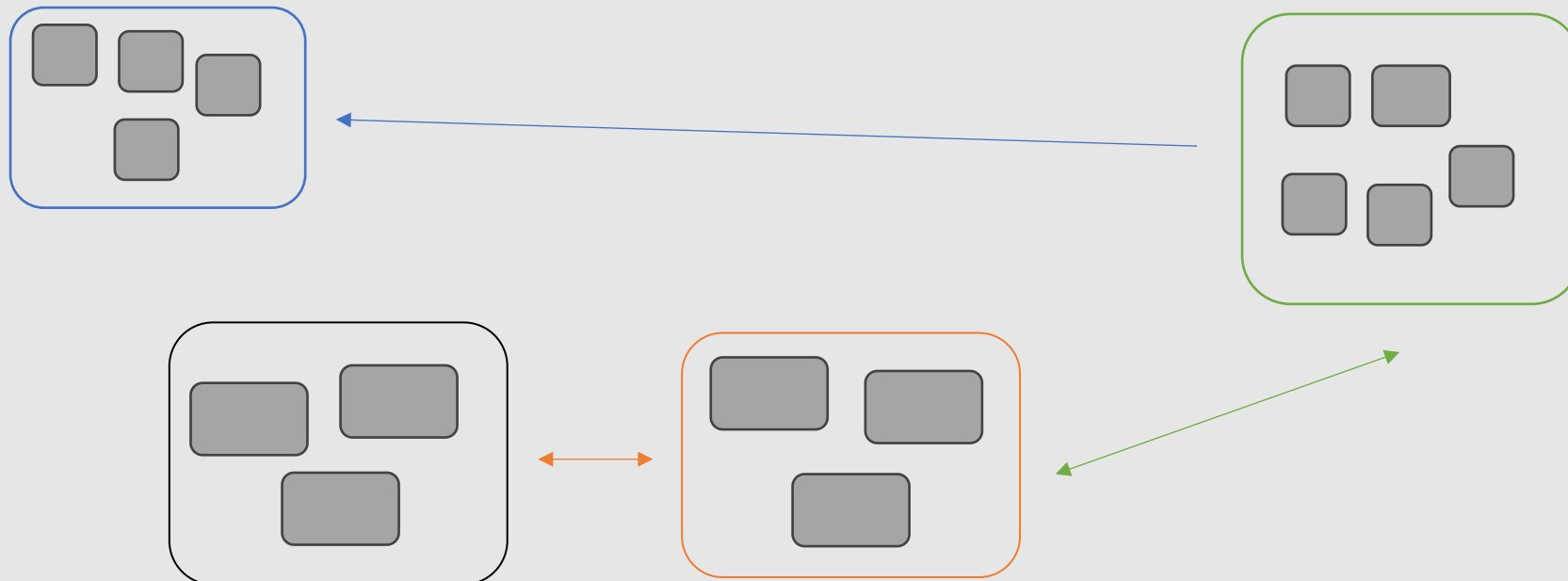
Так что же принесли нам микросервисы

- Те задачи, которые были раньше на всю компанию, они ушли на уровень ниже



Так что же принесли нам микросервисы

- Те задачи, которые были раньше на всю компанию, они ушли на уровень ниже
- На уровне solution архитектуры стало сложнее, потому что надо помнить, что у тебя за каждым кубиком скрывается свой мир



Так что же принесли нам микросервисы

- Те задачи, которые были раньше на всю компанию, они ушли на уровень ниже
- На уровне solution архитектуры стало сложнее, потому что надо помнить, что у тебя за каждым кубиком скрывается свой мир
- Произошло это просто потому что системы которые мы разрабатываем стали сложнее, и это нормально

И что с этим делать?

О чём сегодня поговорим

- Что такое системный анализ в разработке
- Интеграция с внешними системами компании
- В чём разница Solution архитектора и Аналитика
- Что поменялось с приходом микросервисами
- **Создание среды помогающей решать задачи**
- Итоги

И что с этим делать?

Учиться, учиться и еще раз
учиться!

© Джейсон Стэтхэм

Какие есть способы


- Документация и фокус на том, чтобы документация была максимально понятной

▼ [servicename] Краткое принятое название, отражаю

- v1 База данных : [servicename]
- v1 Дизайн-схема: [servicename]
- ▼ Описание API: [servicename]
 - v1 METHOD1 [servicename]: Что делаем с какой сущ
 - v1 METHOD 2 [servicename]: Что делаем с какой сущ

[servicename] Краткое принятое название, отражающее функционал

Создатель Шпер Наталья Владимировна, отредактировано июл 31, 2023 [+ Add Approval](#)

Описание	Пара фраз о том зачем нам этот сервис, какова его зона ответственности
Команда	 Platform Team: переиспользуемые платформенные решения
Статус	DEV
Мейнтейнер	ФИО1, ФИО2
Аналитик	Шпер Наташа
URL	https://{host}/api/v1/servicename
Репозиторий	{repName}
DB	baseName

Какие есть способы

- PullRequest (Шаринг знаний через Approve документации)

Title *

Approvers *

Notifications


- Send Approval Requests
- Receive Status Updates

Expiry

- Expire on Edit
- Expire on Date

Other

- Set Required Approvals



Start by adding a Team or User

Какие есть способы

- Meetup
 - Делимся новыми подходами
 - Оставляем запись для последующего просмотра
 - Обсуждаем улучшения и изменения

Какие есть способы

- Onboarding
 - Подходы в проектировании
 - Работа системы и устройство доменной области
 - Общие правила для контрактов, формирование ошибок, фильтрация
 - Задачи на понимание
 - Ментор, который помогает разобраться

Какие есть способы

- Шаблоны для создания аналитики

Создать описание БД

Создать дизайн-схему

Создать общую страницу методов

Создать описание конфигов

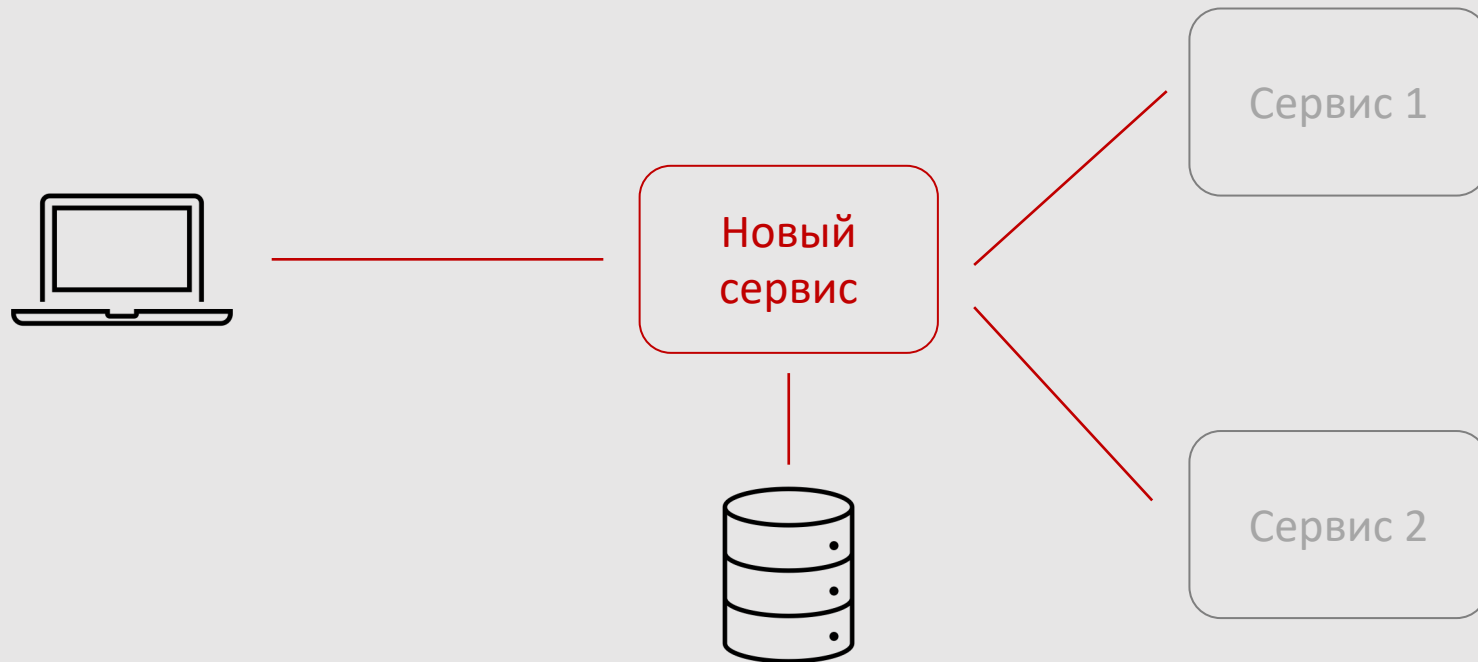
Создать описание ИТ

Какие есть способы

- Общие встречи по принятию решений
 - Раз в неделю встреча гильдий или направления компетенции
 - Раз в неделю встреча лидеров
 - Раз в две недели встреча Head of гильдии

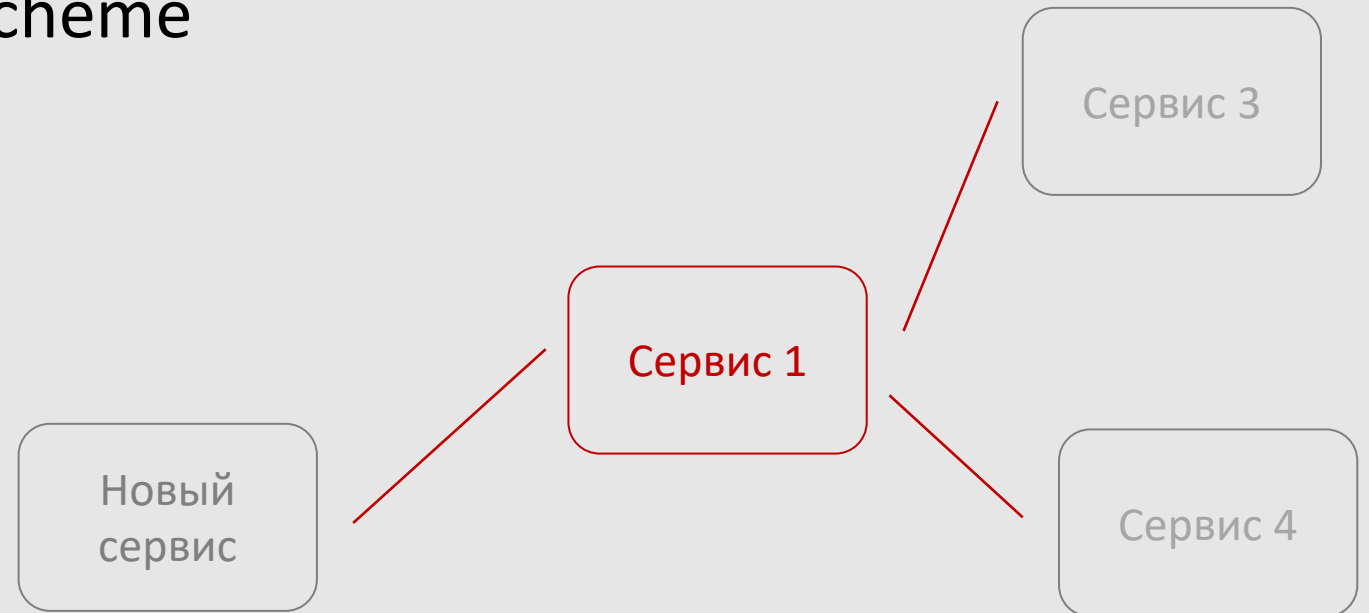
Какие есть способы

- Личный lifehack - Design scheme



Какие есть способы

- Личный lifehack - Design scheme



Что же мы сделали?

- Погрузили команду в контекст предметной области

Что же мы сделали?

- Погрузили команду в контекст предметной области
- Решили задачу sharing'a знаний

Что же мы сделали?

- Погрузили команду в контекст предметной области
- Решили задачу sharing'a знаний
- Дали инструменты для упрощения работы

Что же мы сделали?

- Погрузили команду в контекст предметной области
- Решили задачу sharing'a знаний
- Дали инструменты для упрощения работы
- Мотивация

О чём сегодня поговорим

- Что такое системный анализ в разработке
- Интеграция с внешними системами компании
- В чём разница Solution архитектора и Аналитика
- Что поменялось с приходом микросервисами
- Создание среды помогающей решать задачи
- **Итоги**

Итоги

- Аналитики стали на равных разговаривать с solution архитекторами (кто-то им стал=))
- Улучшилось понимание всего ИТ - ландшафта проекта
- Информация лежит не только в одной голове
- Качество проработки решений выросло
- Появилось классное комьюнити куда ты можешь прийти за советом

Итоги

- Аналитики стали на равных разговаривать с solution архитекторами (кто-то им стал=))
- Улучшилось понимание всего ИТ - ландшафта проекта
- Информация лежит не только в одной голове
- Качество проработки решений выросло
- Появилось классное комьюнити куда ты можешь прийти за советом

Итоги

- Аналитики стали на равных разговаривать с solution архитекторами (кто-то им стал=))
- Улучшилось понимание всего ИТ - ландшафта проекта
- Информация лежит не только в одной голове
- Качество проработки решений выросло
- Появилось классное комьюнити куда ты можешь прийти за советом

Итоги

- Аналитики стали на равных разговаривать с solution архитекторами (кто-то им стал=))
- Улучшилось понимание всего ИТ - ландшафта проекта
- Информация лежит не только в одной голове
- Качество проработки решений выросло
- Появилось классное комьюнити куда ты можешь прийти за советом

Итоги

- Аналитики стали на равных разговаривать с solution архитекторами (кто-то им стал=))
- Улучшилось понимание всего ИТ - ландшафта проекта
- Информация лежит не только в одной голове
- Качество проработки решений выросло
- Появилось классное комьюнити куда ты можешь прийти за советом

Что же из нас делает Solution'ов

- Желание разобраться в том как всё работает
- Грамотно построенная среда

Вопросы

- [facebook.com/chernukhin.maksim](https://www.facebook.com/chernukhin.maksim)
- TG @MaksCher

