# История трансформации:
## как мы не справились с 20 000+ RPS, и что из этого вынесли

**Немировский Лев**

Руководитель направления, ПСБ

HL 2024 HighLoad++

# Начальные условия



**Проект:** Международная образовательная платформа «Содружество»

**Ключевой функционал:** проведение международной олимпиады по финансовой безопасности

**Архитектура:** сервис-ориентированная

**Требования:**

- 500 000 DAU → 300 RPS
- Целевая производительность: 1000 RPS
- Время загрузки < 4 секунды

**Сроки:** 3 месяца на MVP

HighLoad++ 2024

# Компромиссы и технический долг



## Сжатые сроки → Быстрые решения

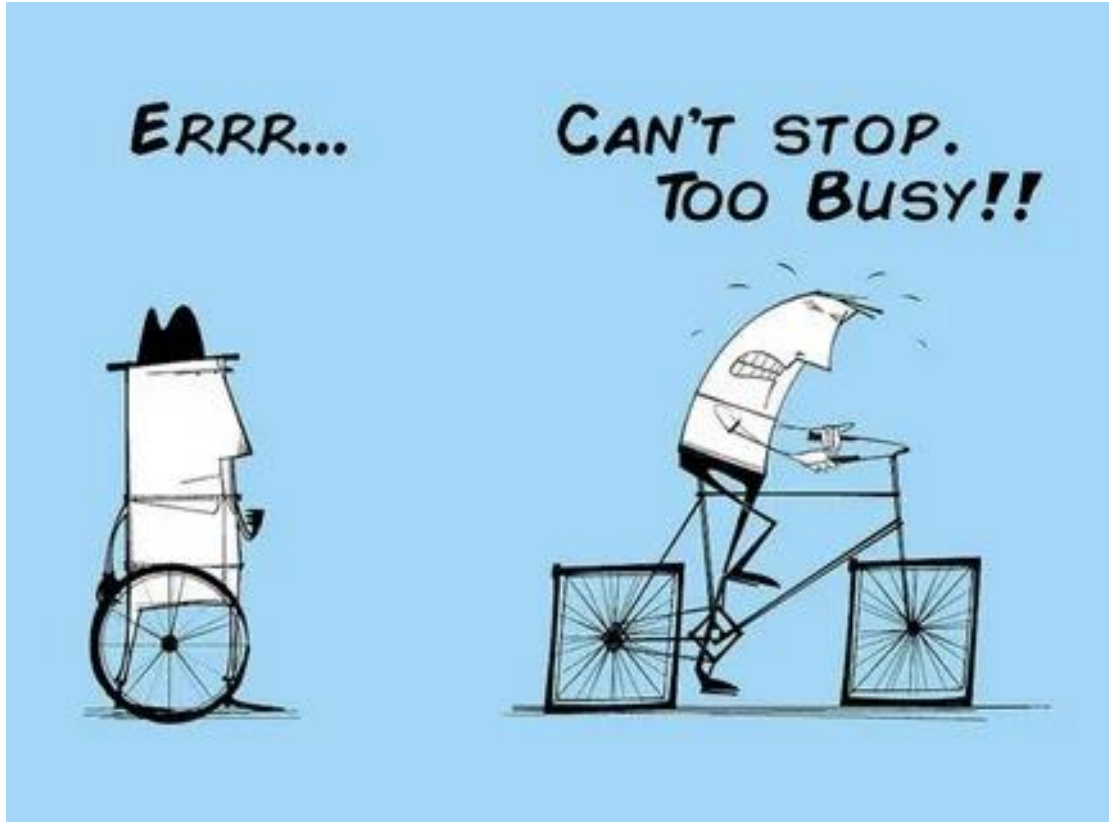
- Отложенный мониторинг
- Отложенное логирование
- Часть функционала — коробка на Ruby
- Вторая часть — на Python вместо Ruby подрядчиком


HighLoad++ 2024

# Первые тревожные звоночки



- Переезд перед запуском
- Нагрузочное тестирование:
  - Внутренний контур: 200 RPS
  - Облако: 700 RPS
  - Целевые показатели: 1000 RPS




HighLoad++ 2024

# Первая победа и её цена



**Цель:** проведение финала международной олимпиады

**Тактическое решение:** разнесение публикации результатов по времени

**Результат:** система выстояла

**Цена:**

- Спидкодинг ночью
- Стресс команды
- Растущий технический долг

# Новая цель:

проведение международной олимпиады
полностью онлайн

# Затишье перед бурей



- ✗ Полный переход на Ruby
- ✗ Нормальный мониторинг
- ✗ Профессиональное нагрузочное тестирование
- ✗ Отличные метрики
- ✗ Полная команда

Стресс команды

# Час X - 20 000+ RPS



Неожиданный наплыв пользователей

Неожиданное поведение — эффект F5

Полный отказ системы


Отсутствие инструментов диагностики

HighLoad++ 2024

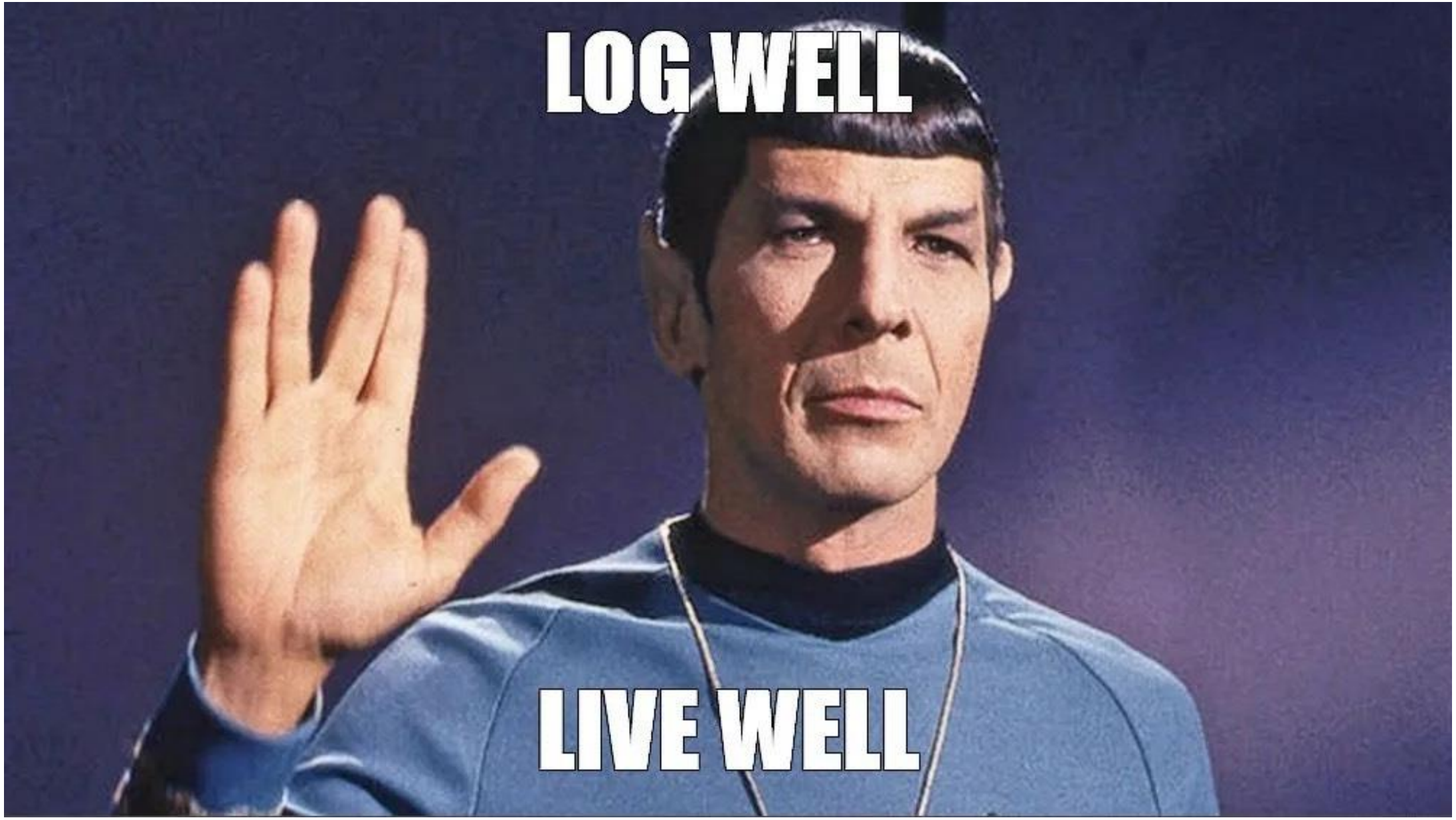
# Антикризисные меры



- Организация процесса:
  - Централизованные статусы
  - Единый канал коммуникации с пользователями — email
- Деление команды на Dev и Support
- Распределение олимпиады по дням
- Реализация мониторинга

HighLoad++ 2024

# Поехали рефакторить




HighLoad++ 2024

# Оптимизация запросов БД

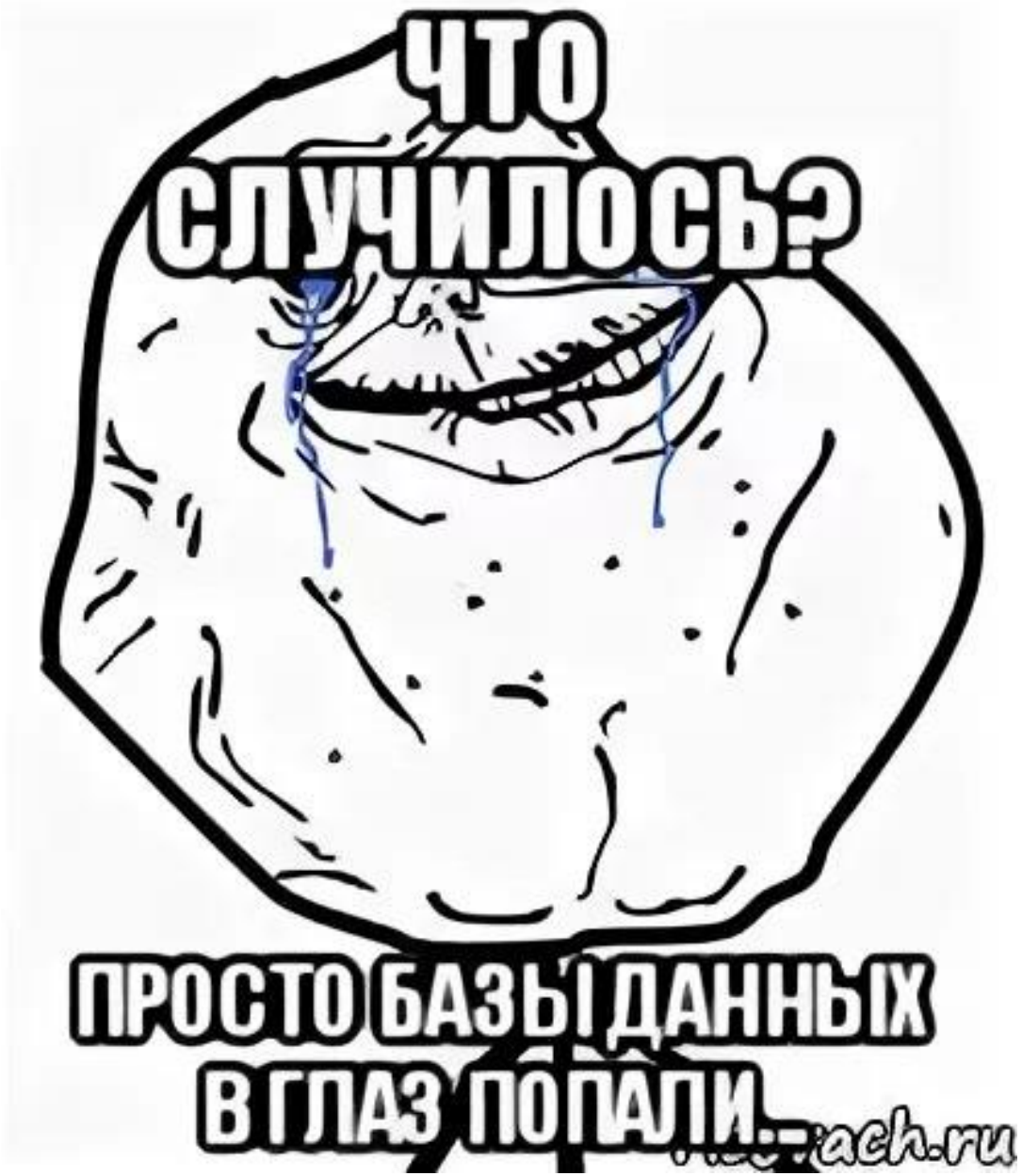


Untitled-1

```
-- Было
SELECT lower(email), ... FROM users;
--Стало
REATE INDEX users_email_lower_idx ON users (lower(email));
```

HighLoad++ 2024

# Мастер БД



# Тех.долг API

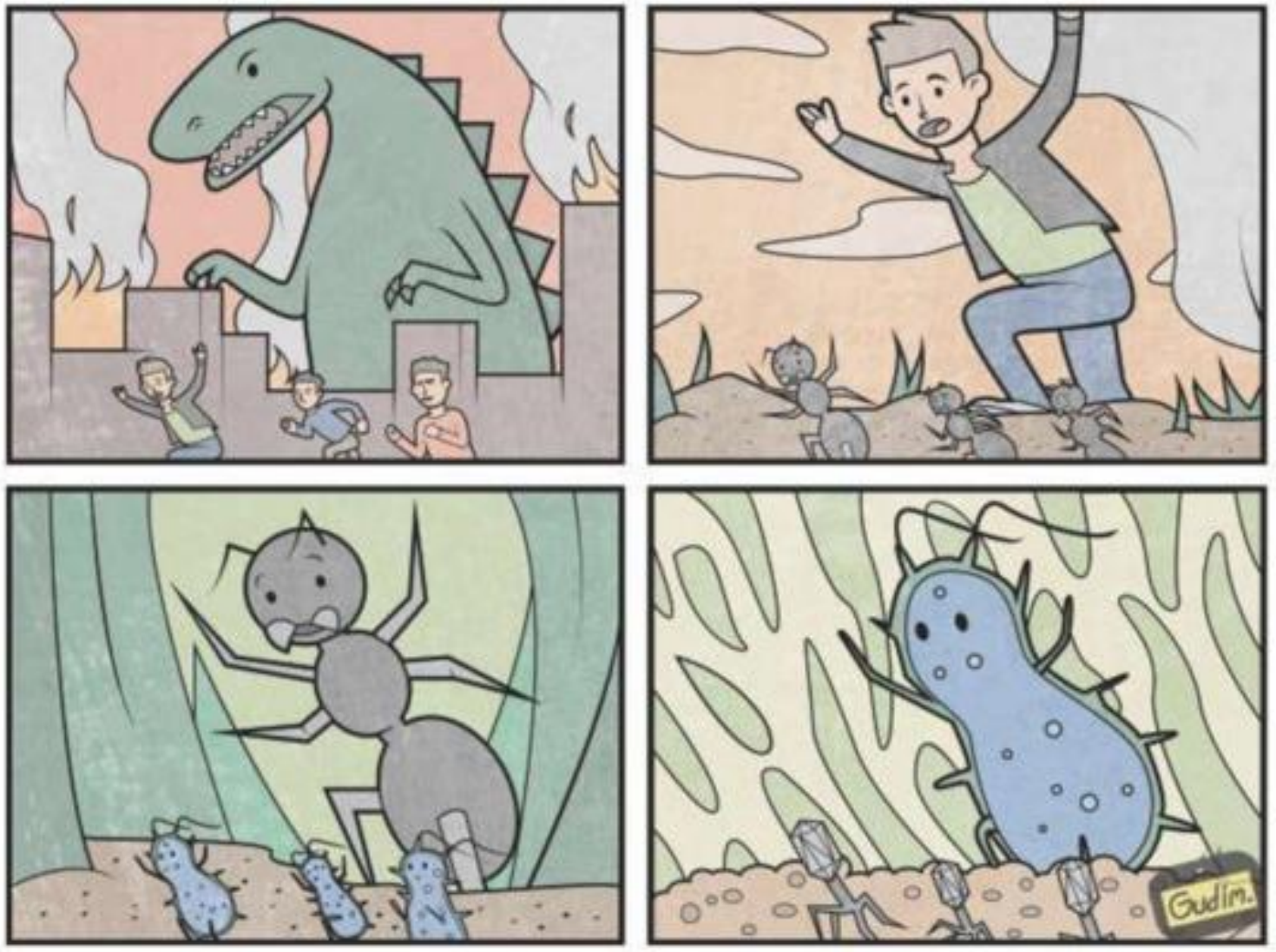


ru

HighLoad++ 2024

# Масштабирование подов






HighLoad++ 2024

# За ~неделю



- Стали выдерживать ~10 000 RPS
- Получили цель в 30 000 RPS
- Ключевые улучшения:
  - Мониторинг
  - Оптимизация БД (вычистили самые тяжелые запросы)
  - По API быстро, где могли, в бэклог, где без вариантов
  - Масштабирование инфраструктуры
- Получили карт-бланш

HighLoad++ 2024

# Результаты проекта



- ~170 000 пользователей
- 70 стран
- Более 90 000 тысяч — участники IV Международной олимпиады по финансовой безопасности
- Вся олимпиада проведена онлайн
- Платформа стала технологическим партнером «Международного движения по финансовой безопасности»
- Огромное количество благодарностей и планов

HighLoad++ 2024

# DevOps



# DevOps и мониторинг



- API Gateway + OpenTelemetry
- Базовый мониторинг (slow_log, ресурсы)
- Метрики/трейсинг через готовые решения
- Алертинг
- Проверить, что логи пишутся

HighLoad++ 2024

# Планирование нагрузки



- Проактивное планирование
  - Механизм распределения нагрузки
  - Ручное распределение, если механизм не успел или отказал
  - Система оповещений (не только email)!!!
- А еще…

HighLoad++ 2024

# Расчет нагрузки



Untitled-1

```
# Расчет
500000 DAU * 50 Request / 86400 = 300 * 3 = ~1 000 RPS
# Реальность
25 000 человек пришло на олимпиаду в 09:58 = ~ 20 000 RPS
```

- Факторы пиковой нагрузки:
  - Время публикации результатов
  - Время начала олимпиады
  - Время окончания олимпиады
- Географическое распределение пользователей
- F5-эффект и его последствия

HighLoad++ 2024

# Масштабирование и отказоустойчивость



- Тестирование разного количества подов
- Проверка критической инфраструктуры:
  - Верификация заявленных возможностей
  - Документирование реального состояния
- План действий при отказах


HighLoad++ 2024

# Работа с ограничениями и техническим долгом



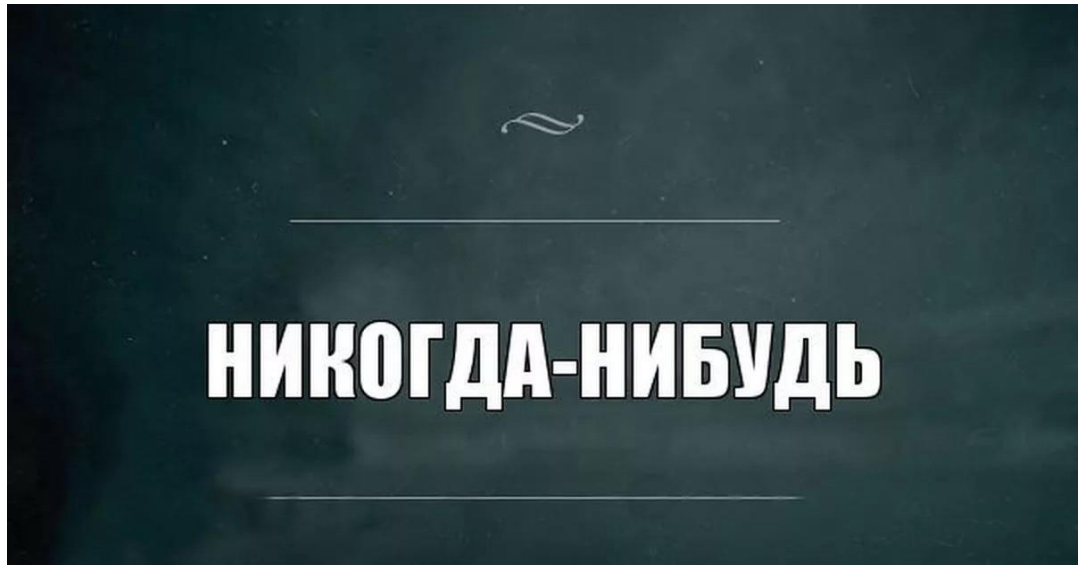
- Осознанные компромиссы:
  - Четкие критерии выбора
  - Оценка последствий
  - План улучшений
- Приоритизация ресурсов
- Документирование технического долга


HighLoad++ 2024

# Управление разработкой



- "Потом" не наступает:
  - Технический долг = продуктовые фичи
  - Ресурсы на инфраструктуру сразу
- Честная оценка рисков
- Открытое обсуждение последствий




HighLoad++ 2024

# Управление разработкой



## План Б до часа X

- Механизмы деградации функционала
- Сценарии распределения нагрузки

## Баланс скорости и надёжности

- MVP ≠ грязный код
- Выделение критичных компонентов
- Техническая экспертиза при проектировании и согласовании фич

HighLoad++ 2024

# Мы тоже стали лучше



- Внедрили правило 80/20: фиксированная квота на технический долг в каждом спринте
- Перешли от авралов к регулярному рефакторингу и улучшению инфраструктуры
- Внедрили процессы DevOps
- Ввели архитектурную экспертизу и документирование всех технических решений
- Внедрили культуру надёжности: план Б и оценка рисков для каждого решения

HighLoad++ 2024

В хорошем оркестре недостаточно лучших музыкантов и инструментов.

Без нот и дирижера музыки не будет.

HighLoad++ 2024

# Спасибо за внимание

# QR голосования


Немировский Лев

Руководитель направления, ПСБ

@bodrcoder

levn.me


ПСБ
ЦИФРОВАЯ
ЛАБОРАТОРИЯ

HL 2024 HighLoad++