

ПРОПУСК ЗА КУЛИСЫ

О КАКИХ ЗАЯВКАХ МЕЧТАЕТ ПРОГРАММНЫЙ КОМИТЕТ?

на примере конференции HolyJS

КОГДА ПОРА БЫТЬ СПИКЕРОМ?

- Реализован
уникальный проект
(есть прод опыт)

КОГДА ПОРА БЫТЬ СПИКЕРОМ?

- Реализован
уникальный проект
(есть прод опыт)
- Есть уникальный
опыт (сочетание
разных областей,
сфер)

КОГДА ПОРА БЫТЬ СПИКЕРОМ?

- Реализован уникальный проект (есть прод опыт)
- Есть уникальный опыт (сочетание разных областей, сфер)
- Эксперт в какой-либо теме

КОГДА ПОРА БЫТЬ СПИКЕРОМ?

- Реализован уникальный проект (есть прод опыт)
- Есть уникальный опыт (сочетание разных областей, сфер)
- Эксперт в какой-либо теме
- Хорошие speaker-skills

КОГДА ПОРА БЫТЬ СПИКЕРОМ?

- Реализован уникальный проект (есть прод опыт)
- Есть уникальный опыт (сочетание разных областей, сфер)
- Эксперт в какой-либо теме
- Хорошие speaker-skills
- Зовут поделиться экспертизой

КОГДА ПОРА БЫТЬ СПИКЕРОМ?

- Реализован уникальный проект (есть прод опыт)
- Есть уникальный опыт (сочетание разных областей, сфер)
- Эксперт в какой-либо теме
- Хорошие speaker-skills
- Зовут поделиться экспертизой
- Есть желание выступить

а судьи кто?



Программный комитет собирается из N-числа экспертов по всем направлениям конференции



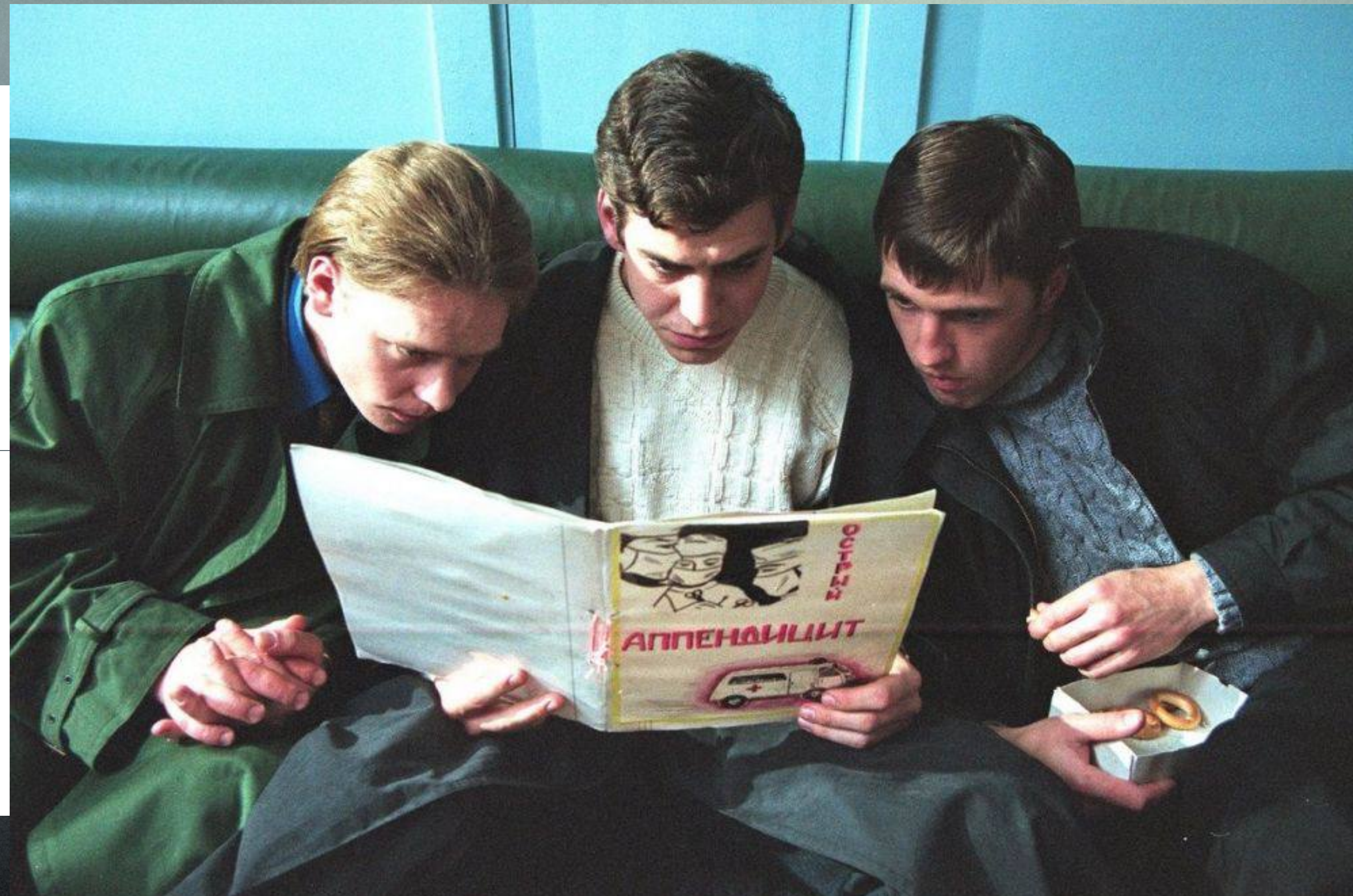
Программный комитет собирается из N-числа экспертов по всем направлениям конференции

КАК СОБИРАЕТСЯ ПК?



ЭКСПЕРТНОСТЬ

КАК СОБИРАЕТСЯ ПК?



НАСМОТРЕННОСТЬ

КАК СОБИРАЕТСЯ ПК?



МОТИВАЦИЯ

КАК СОБИРАЕТСЯ ПК?



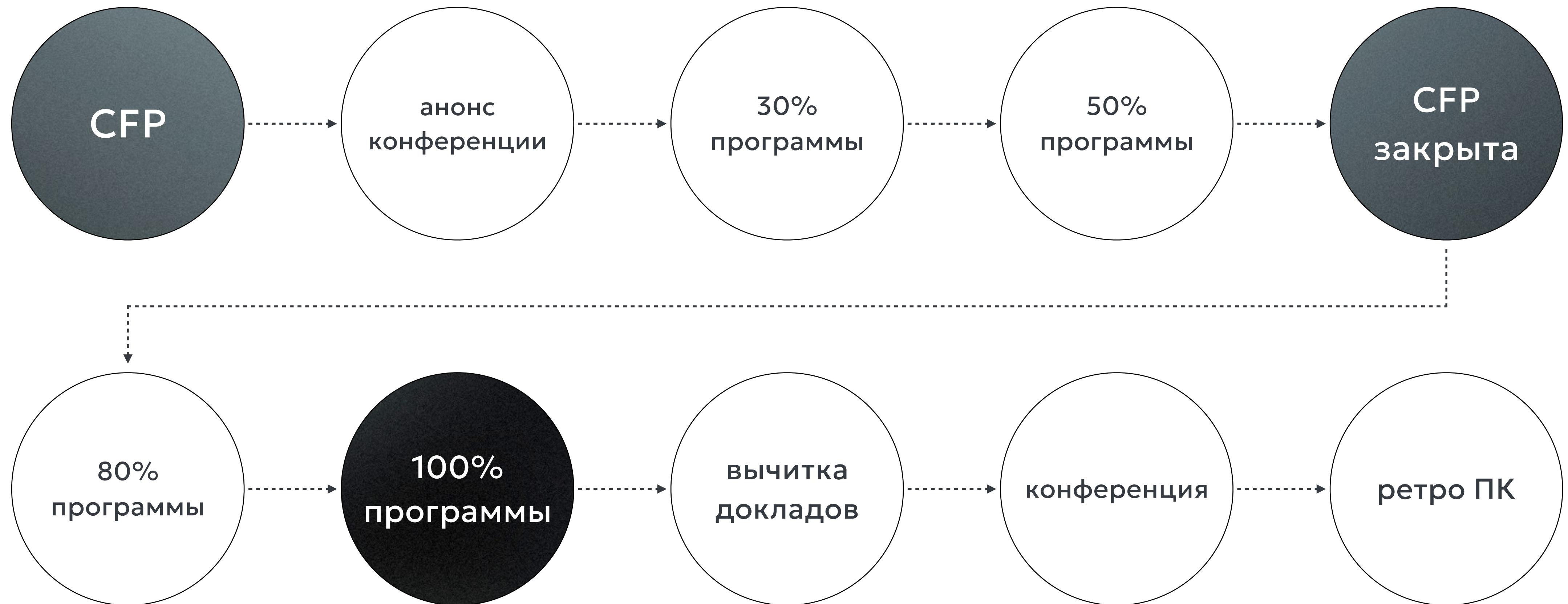
КОММУНИКАбельНОСТЬ

РОЛИ В ПК

- Приглашения
- Сорсинг спикеров
- Первичные созвоны
- Ревью докладов
- Кураторство
- Помощь маркетингу
- Участие в доп активностях
- Ревью партнерских активностей
- Взаимодействие с комьюнити

ФОРМИРОВАНИЕ ПРОГРАММЫ

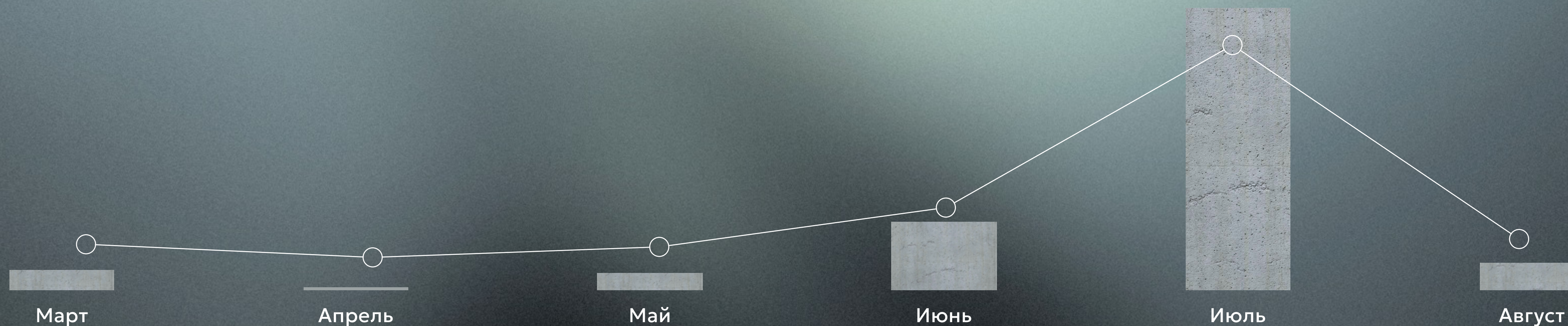
МАЙЛСТОУНЫ



КАК ПРИХОДЯТ
ЗАЯВКИ?

РАСПРЕДЕЛЕНИЕ ЗАЯВОК

НА ПРИМЕРЕ СЕЗОНА 2023 ОСЕНЬ



- Март-апрель — подготовка HolyJS 2023 Spring
- Май — анонс CFP на закрытии конференции
- Июнь — пора отпусков, активный сбор заявок
- Июль — окончание приема заявок
- Август — опоздавшие, точечный донабор

Как мы чувствовали себя в июле?

НА ПРИМЕРЕ СЕЗОНА 2023 ОСЕНЬ



- Март-апрель — подготовка HolyJS 2023 Spring
- Май — анонс CFP на закрытии конференции
- Июнь — пора отпусков, активный сбор заявок
- Июль — окончание приема заявок
- Август — опоздавшие, точечный донабор

Как мы чувствовали себя в июле?



Март

Август

- Март-апрель
- Май — анонс
- Июнь — пора

донабор

ОБЩЕНИЕ СО СПИКЕРОМ

1 КОНТАКТ

ПИСЬМО ОТ КООРДИНАТОРА

**Начали рассматривать
заявку — отправляем в
MatterMost сообщение
о процессе**



ОБЩЕНИЕ СО СПИКЕРом



ОБСУЖДЕНИЕ
ПЕРВИЧНЫЙ СОЗВОН

На первичном созвоне
с куратором
обсуждается план
доклада + наработки

ОБЩЕНИЕ СО СПИКЕРОМ

ПАУЗА

РАССМОТРЕНИЕ ЗАЯВОК

Сразу решение не
принимаем:
запрашиваем доп
материалы и заявка
проходит ревью внутри
ПК



ОБЩЕНИЕ СО СПИКЕРОМ



ОТВЕТ

ФИНАЛЬНОЕ РЕШЕНИЕ

Через три недели
после закрытия CFR —
крайний срок для
принятия решения

На что обращаем
внимание?

УНИКАЛЬНОСТЬ

- ЗАСВЕТИЛСЯ ЛИ ДОКЛАД НА МЕРОПРИЯТИЯХ?
- БЫЛИ ЛИ ПОХОЖИЕ ДОКЛАДЫ У ДРУГИХ СПИКЕРОВ?

СПΙΚΕΡСКИЕ НАВЫКИ

- ЕСТЬ ЛИ У СПИКЕРА ОПЫТ ВЫСТУПЛЕНИЙ
- КАКАЯ РЕАКЦИЯ НА ФИДБЕК?
- КАК СПИКЕР РЕАГИРУЕТ НА КАМЕРУ/АУДИТОРИЮ?
- КАК СПИКЕР ОТВЕЧАЕТ НА ВОПРОСЫ И ФОРМУЛИРУЕТ ТЕЗИСЫ

актуальность

- **НАСКОЛЬКО
КОНТЕНТ ПОЛЕЗЕН
СЕЙЧАС НАШЕЙ
ЦА?**

- **БУДЕТ ЛИ
ПОЛЕЗЕН В
БУДУЩЕМ?**

ЭКСПЕРТНОСТЬ

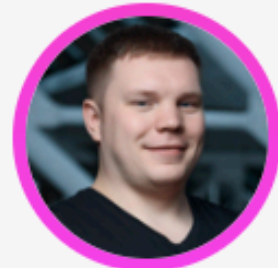
- ДЕЛАЛ ЛИ РУКАМИ СПИКЕР ТО, ЧТО ОПИСЫВАЕТ?
- НАСКОЛЬКО ГЛУБОКО ИЗУЧЕН МАТЕРИАЛ?

ПРИМЕНИМОСТЬ

- ОПИСАНЫ ЛИ ВОЗМОЖНЫЕ ПРОБЛЕМЫ?
- ЕСТЬ ЛИ РЕШЕНИЕ ДЛЯ ЭТИХ ПРОБЛЕМ?

КАК ВЫБИРАЕМ?

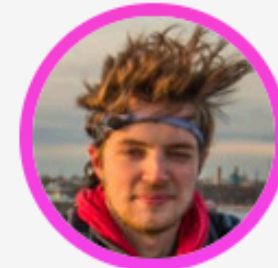
Программный комитет



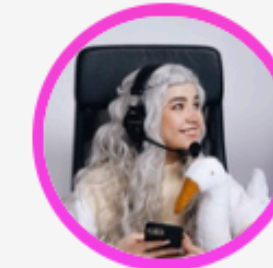
Алексей Золотых
МойОфис



Никита Сидоров
Яндекс Маркет



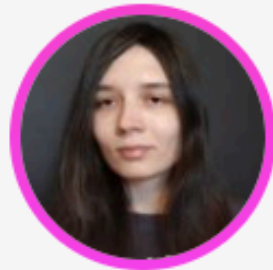
Александр Коротаев



Дана Стоянова
JUG Ru Group



Иван Попельшев
White Frame



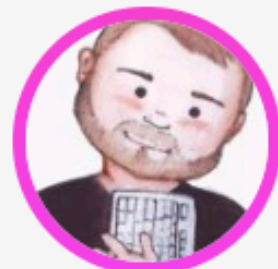
Полина Гуртовая
Recraft



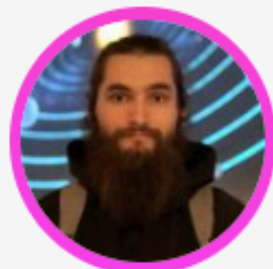
Ярослав Лосев
Яндекс Путешествия



Виктор Хомяков



Василий Ванчук



Семён Левенсон
Яндекс Маркет

«КВОРУМ ТРЕХ»

БИТВА ЗА ПРОГРАММУ



БИТВА ЗА ПРОГРАММУ



ДОП РЕВЬЮ

БИТВА ЗА ПРОГРАММУ

	Семён	Никита	Саша	Вася	Ваня	Яросла	Виктор	Лёша		
	5	5	4	2	3	4	0	4	3,375	27
	2	3	2	3	0	5	4	0	2,375	19
	0	4	0	0	4	0	5	5	2,25	18
	0	0	5	4	2	1	3	0	1,875	15
	4	0	3	0	0	3	0	0	1,25	10
	0	0	1	0	5	0	2	0	1	8
	3	0	0	0	1	0	0	3	0,875	7
	1	0	0	1	0	0	1	2	0,625	5
	3	1	0	0	0	0	0	0	0,5	4
	0	2	0	0	0	2	0	0	0,5	4
	0	0	0	5	0	0	0	1	0,75	6
	0	0	0	0	0	0	0	0	0	0

ФИНАЛЬНОЕ ГОЛОСОВАНИЕ


БИТВА ЗА ПРОГРАММУ

		Уже в программе	Заявки	Кого звать
Блоки				
Внутреннее устройство программных решений				
Ничего из внутреннего устройства программных решений	9,52 % 4			
Языки описания веб-сервисов (SOAP/WSDL, OpenAPI, GraphQL)	52,38 % 22			
Языки обмена сообщениями (XML, JSON/YAML)	45,24 % 19			
Работа с интеграционным API (Swagger, Postman, cURL, HTTPie, Insomnia)	57,14 % 24			
Очереди, шины данных и брокеры сообщений (RabbitMQ, Kafka, Spark)	66,67 % 28			
DSL и Low-code платформы	26,19 % 11			
	4,76 % 2			
Практика TOGAF, O-AAF, ETL				
Требования				
Ничего из требований	2,38 % 1			
Описание и представление (user story, use case, JTBD, CJM)	69,05 % 29			
Нефункциональные требования (производительность, capacity management, compliance, информационная безопасность)	52,38 % 22			
UX/UI	35,71 % 15			
Инструменты, подходы и стандарты (ISO 29148, ГОСТ 34, ГОСТ 19)	26,19 % 11			

АНАЛИЗ НАПРАВЛЕНИЙ

ЧТО ТАКОЕ «ХОРОШАЯ»
ЗАЯВКА?

«Хорошая» заявка

Название доклада:	Пока не знаю
Название доклада EN:	Пока не знаю
Язык доклада:	
Описание доклада:	Next js
Описание доклада EN:	Next js

«Хорошая» заявка

Название доклада Потокус — новый подход к созданию бессмертного интерактивного контента

EN:

Язык доклада:

RU

Описание доклада:

- ✓ Потокус — это технология, которая превращает разрозненные медиа в полноценный контент и собирает бесконечный интерактивный эфир. При этом:
 - контент не запечатывается в файлы, а собирается плеером на лету; это позволяет делать «бессмертный контент», обновляя алгоритмы сборки контента: переходы, эффекты и другое;
 - написан движок для виджетов — интерактивных элементов поверх трансляции, которые позволяют аудитории взаимодействовать с эфиром и подгружать динамические данные из разных источников;
 - написан локальный фреймворк для виджетов, который позволяет дописывать их под потребности клиентов отдельно от основного продукта;
 - создана админка, которая позволяет очень просто собирать контент на таймлайне и составлять расписание из блоков.

Потокус достаточно сложно устроен внутри — пришлось проектировать с нуля всякие инженерные штуки и придумывать то, что ещё никто никогда не делал. Готовых решений в этой области нет, и поэтому проект полон смекалки.

Nuxt, Vue (админка)

pixi.js (плеер)

Целевая аудитория — фронтенд-разработчики, которым интересна не просто вёрстка и стили, а полноценная инженерная разработка на фронте.

«Хорошая» заявка

Потокус — это технология, которая превращает разрозненные медиа в полноценный контент и собирает бесконечный интерактивный эфир. При этом:

- контент не запечатывается в файлы, а собирается плеером на лету; это позволяет делать «бессмертный контент», обновляя алгоритмы сборки контента: переходы, эффекты и другое;
- написан движок для виджетов — интерактивных элементов поверх трансляции, которые позволяют аудитории взаимодействовать с эфиром и подгружать динамические данные из разных источников;
- написан локальный фреймворк для виджетов, который позволяет дописывать их под потребности клиентов отдельно от основного продукта;
- создана админка, которая позволяет очень просто собирать контент на таймлайне и составлять расписание из блоков.

«Хорошая» заявка

Потокус — это технология, которая превращает разрозненные медиа в полноценный контент и собирает бесконечный интерактивный эфир. При этом:

- контент не запечатывается в файлы, а собирается плеером на лету; это позволяет делать «бессмертный контент», обновляя алгоритмы сборки контента: переходы, эффекты и другое;
- написан движок для виджетов — интерактивных элементов поверх трансляции, которые позволяют аудитории взаимодействовать с эфиром и подгружать динамические данные из разных источников;
- написан локальный фреймворк для виджетов, который позволяет дописывать их под потребности клиентов отдельно от основного продукта;
- создана админка, которая позволяет очень просто собирать контент на таймлайне и составлять расписание из блоков.

РАССКАЗ ПРО ПРОДУКТ И
ЕГО УНИКАЛЬНОСТЬ

«Хорошая» заявка

Потокус — это технология, которая превращает разрозненные медиа в полноценный контент и собирает бесконечный интерактивный эфир. При этом:

- контент не запечатывается в файлы, а собирается плеером на лету; это позволяет делать «бессмертный контент», обновляя алгоритмы сборки контента: переходы, эффекты и другое;
- написан движок для виджетов — интерактивных элементов поверх трансляции, которые позволяют аудитории взаимодействовать с эфиром и подгружать динамические данные из разных источников;
- написан локальный фреймворк для виджетов, который позволяет дописывать их под потребности клиентов отдельно от основного продукта;
- создана админка, которая позволяет очень просто собирать контент на таймлайне и составлять расписание из блоков.

РАССКАЗ ПРО ПРОДУКТ И
ЕГО УНИКАЛЬНОСТЬ

Потокус достаточно сложно устроен внутри — пришлось проектировать с нуля всякие инженерные штуки и придумывать то, что ещё никто никогда не делал. Готовых решений в этой области нет, и поэтому проект полон смекалки.

Nuxt, Vue (админка)

pixi.js (плеер)

«Хорошая» заявка

Потокус — это технология, которая превращает разрозненные медиа в полноценный контент и собирает бесконечный интерактивный эфир. При этом:

- контент не запечатывается в файлы, а собирается плеером на лету; это позволяет делать «бессмертный контент», обновляя алгоритмы сборки контента: переходы, эффекты и другое;
- написан движок для виджетов — интерактивных элементов поверх трансляции, которые позволяют аудитории взаимодействовать с эфиром и подгружать динамические данные из разных источников;
- написан локальный фреймворк для виджетов, который позволяет дописывать их под потребности клиентов отдельно от основного продукта;
- создана админка, которая позволяет очень просто собирать контент на таймлайне и составлять расписание из блоков.

РАССКАЗ ПРО ПРОДУКТ И
ЕГО УНИКАЛЬНОСТЬ

Потокус достаточно сложно устроен внутри — пришлось проектировать с нуля всякие инженерные штуки и придумывать то, что ещё никто никогда не делал. Готовых решений в этой области нет, и поэтому проект полон смекалки.

Nuxt, Vue (админка)
pixi.js (плеер)

СЛОЖНОСТИ, С
КОТОРЫМИ СТОЛКНУЛИСЬ
+ НАМЕК НА РЕШЕНИЯ

«Хорошая» заявка

Название доклада

Замена фона в видеозвонке

EN:

Язык доклада:

RU

Описание доклада:

✓ Расскажу как разобрать видео на кадры, найти в кадрах людей, заменить фон и собрать обратно. И еще немного про нейронки, стримы и канвасы.

Какие есть видеоэффекты в VK звонках и как они работают.

Используемые технологии: WebRTC, Insertable Streams, Media Source Extensions, Canvas.

Доклад будет полезен, если вы работаете с видео, звонками или конференциями, делаете свой видеоплеер, сервис групповых звонков или DRM.

«Хорошая» заявка

Название доклада

Замена фона в видеозвонке

EN:

Язык доклада:

RU

Описание доклада:

✓ Расскажу как разобрать видео на кадры, найти в кадрах людей, заменить фон и собрать обратно. И еще немного про нейронки, стримы и канвасы.

Какие есть видеоэффекты в VK звонках и как они работают.

Используемые технологии: WebRTC, Insertable Streams, Media Source Extensions, Canvas.

Доклад будет полезен, если вы работаете с видео, звонками или конференциями, делаете свой видеоплеер, сервис групповых звонков или DRM.

ИНТЕРЕСНАЯ ЗАДАЧА +
ТЕХНОЛОГИИ +
ПРИМЕНИМОСТЬ

«Хорошая» заявка

Название доклада
EN:

Замена фона в видеозвонке

Язык доклада:

RU

Описание доклада:

- ✓ Расскажу как разобрать видео на кадры, найти в кадрах людей, заменить фон и собрать обратно. И еще немного про нейронки, стримы и канвасы. Какие есть видеоэффекты в VK звонках и как они работают.

Используемые технологии: WebRTC, Insertable Streams, Media Source Extensions, Canvas.

Доклад будет полезен, если вы работаете с видео, звонками или конференциями, делаете свой видеоплеер, сервис групповых звонков или DRM.

Детальный план
доклада:

- ✓ 1. Коротко о том, как устроены звонки.
- 2. Концепция замены фона в звонке.
- 3. Разбираем видео на кадры с помощью Insertable Streams.
- 4. Какие есть эффекты в VK звонках и чем они отличаются.
- 5. Наложение overlay эффекта на кадр.
- 6. Немного о нейросетях и определении человека в кадре.
- 7. Замена фона в кадре изображением.
- 8. Немного про blur в целом и blur фона в кадре.
- 9. Сборка стрима из кадров обратно.
- 10. Что еще можно сделать с видео.

ИНТЕРЕСНАЯ ЗАДАЧА +
ТЕХНОЛОГИИ +
ПРИМЕНИМОСТЬ

«Хорошая» заявка

Название доклада
EN:

Замена фона в видеозвонке

Язык доклада:

RU

Описание доклада:

- ✓ Расскажу как разобрать видео на кадры, найти в кадрах людей, заменить фон и собрать обратно. И еще немного про нейронки, стримы и канвасы. Какие есть видеоэффекты в VK звонках и как они работают.

Используемые технологии: WebRTC, Insertable Streams, Media Source Extensions, Canvas.

Доклад будет полезен, если вы работаете с видео, звонками или конференциями, делаете свой видеоплеер, сервис групповых звонков или DRM.

Детальный план
доклада:

- ✓ 1. Коротко о том, как устроены звонки.
- 2. Концепция замены фона в звонке.
- 3. Разбираем видео на кадры с помощью Insertable Streams.
- 4. Какие есть эффекты в VK звонках и чем они отличаются.
- 5. Наложение overlay эффекта на кадр.
- 6. Немного о нейросетях и определении человека в кадре.
- 7. Замена фона в кадре изображением.
- 8. Немного про blur в целом и blur фона в кадре.
- 9. Сборка стрима из кадров обратно.
- 10. Что еще можно сделать с видео.

ИНТЕРЕСНАЯ ЗАДАЧА +
ТЕХНОЛОГИИ +
ПРИМЕНИМОСТЬ

В ПЛАНЕ — ГЛУБИНА

«Хорошая» заявка

Название доклада
EN:

Написание бенчмарков и профилирование для кода на JS/TS

Язык доклада:

RU

Описание доклада:

✓ JavaScript - богатый язык, в нём большинство задач можно решить несколькими способами. Часто фронтендеры хотят сравнить варианты решения и выбрать самый быстрый, чтобы использовать его в своём проекте. Это важно в клиентском коде, который выполняется в браузере - метрики Web Vitals FID, INP, TBT, TTI как раз про это. Это также важно и в серверном коде - отзывчивость на запросы (TTFB, TTLB) и RPS зависят от скорости выполнения кода в Node.js.

Для оценки скорости работы кода и сравнения скорости разных вариантов служат бенчмарки. Для фиксирования результатов и недеградации по скорости используются performance-тесты. Для оценки скорости работы кода и поиска в нём неоптимальных мест выполняется профилирование.

Я расскажу

- какие бывают ошибки при написании бенчмарков,
- как избежать ошибок,
- как писать хорошие бенчмарки и performance-тесты,
- как профилировать код

«Хорошая» заявка

JavaScript - богатый язык, в нём большинство задач можно решить несколькими способами. Часто фронтендеры хотят сравнить варианты решения и выбрать самый быстрый, чтобы использовать его в своём проекте. Это важно в клиентском коде, который выполняется в браузере - метрики Web Vitals FID, INP, TBT, TTI как раз про это. Это также важно и в серверном коде - отзывчивость на запросы (TTFB, TTLB) и RPS зависят от скорости выполнения кода в Node.js.

«Хорошая» заявка

JavaScript - богатый язык, в нём большинство задач можно решить несколькими способами. Часто фронтендеры хотят сравнить варианты решения и выбрать самый быстрый, чтобы использовать его в своём проекте. Это важно в клиентском коде, который выполняется в браузере - метрики Web Vitals FID, INP, TBT, TTI как раз про это. Это также важно и в серверном коде - отзывчивость на запросы (TTFB, TTLB) и RPS зависят от скорости выполнения кода в Node.js.

ОПРЕДЕЛЕНИЕ ПРОБЛЕМЫ
+ ГЛУБИНА

«Хорошая» заявка

JavaScript - богатый язык, в нём большинство задач можно решить несколькими способами. Часто фронтендеры хотят сравнить варианты решения и выбрать самый быстрый, чтобы использовать его в своём проекте. Это важно в клиентском коде, который выполняется в браузере - метрики Web Vitals FID, INP, TBT, TTI как раз про это. Это также важно и в серверном коде - отзывчивость на запросы (TTFB, TTLB) и RPS зависят от скорости выполнения кода в Node.js.

Для оценки скорости работы кода и сравнения скорости разных вариантов служат бенчмарки. Для фиксирования результатов и недеградации по скорости используются performance-тесты. Для оценки скорости работы кода и поиска в нём неоптимальных мест выполняется профилирование.

ОПРЕДЕЛЕНИЕ ПРОБЛЕМЫ
+ ГЛУБИНА

«Хорошая» заявка

JavaScript - богатый язык, в нём большинство задач можно решить несколькими способами. Часто фронтендеры хотят сравнить варианты решения и выбрать самый быстрый, чтобы использовать его в своём проекте. Это важно в клиентском коде, который выполняется в браузере - метрики Web Vitals FID, INP, TBT, TTI как раз про это. Это также важно и в серверном коде - отзывчивость на запросы (TTFB, TTLB) и RPS зависят от скорости выполнения кода в Node.js.

Для оценки скорости работы кода и сравнения скорости разных вариантов служат бенчмарки. Для фиксирования результатов и недеградации по скорости используются performance-тесты. Для оценки скорости работы кода и поиска в нём неоптимальных мест выполняется профилирование.

ОПРЕДЕЛЕНИЕ ПРОБЛЕМЫ
+ ГЛУБИНА

ВОЗМОЖНОЕ РЕШЕНИЕ

«Хорошая» заявка

JavaScript - богатый язык, в нём большинство задач можно решить несколькими способами. Часто фронтендеры хотят сравнить варианты решения и выбрать самый быстрый, чтобы использовать его в своём проекте. Это важно в клиентском коде, который выполняется в браузере - метрики Web Vitals FID, INP, TBT, TTI как раз про это. Это также важно и в серверном коде - отзывчивость на запросы (TTFB, TTLB) и RPS зависят от скорости выполнения кода в Node.js.

ОПРЕДЕЛЕНИЕ ПРОБЛЕМЫ
+ ГЛУБИНА

Для оценки скорости работы кода и сравнения скорости разных вариантов служат бенчмарки. Для фиксирования результатов и недеградации по скорости используются performance-тесты. Для оценки скорости работы кода и поиска в нём неоптимальных мест выполняется профилирование.

ВОЗМОЖНОЕ РЕШЕНИЕ

Я расскажу

- какие бывают ошибки при написании бенчмарков,
- как избежать ошибок,
- как писать хорошие бенчмарки и performance-тесты,
- как профилировать код

«Хорошая» заявка

JavaScript - богатый язык, в нём большинство задач можно решить несколькими способами. Часто фронтендеры хотят сравнить варианты решения и выбрать самый быстрый, чтобы использовать его в своём проекте. Это важно в клиентском коде, который выполняется в браузере - метрики Web Vitals FID, INP, TBT, TTI как раз про это. Это также важно и в серверном коде - отзывчивость на запросы (TTFB, TTLB) и RPS зависят от скорости выполнения кода в Node.js.

Для оценки скорости работы кода и сравнения скорости разных вариантов служат бенчмарки. Для фиксирования результатов и недеградации по скорости используются performance-тесты. Для оценки скорости работы кода и поиска в нём неоптимальных мест выполняется профилирование.

Я расскажу


- какие бывают ошибки при написании бенчмарков,
- как избежать ошибок,
- как писать хорошие бенчмарки и performance-тесты,
- как профилировать код

ОПРЕДЕЛЕНИЕ ПРОБЛЕМЫ
+ ГЛУБИНА

ВОЗМОЖНОЕ РЕШЕНИЕ

ПЛАН СОБИРАЕТ ПУНКТЫ
ВЫШЕ ВОЕДИНО — TL;DR

«Хорошая» заявка

Название доклада:	Пока не знаю
Название доклада EN:	Пока не знаю
Язык доклада:	
Описание доклада:	Next js
Описание доклада EN:	Next js

ЧТО ДЕЛАТЬ, ЕСЛИ МЫ НЕ
ПОНИМАЕМ ЗАЯВКУ?



ЧТО ДЕЛАТЬ?

- Первичный созвон
- Подробный план (40-50 пунктов)
- Анализ бэкграунда спикера
- Подбор возможных тем
- и так далее

КАК ДОСТУЧАТЬСЯ ДО ПК?



ДО ПОДАЧИ: ПРЯМАЯ ЛИНИЯ

Не знаете о чем рассказать? Мы готовы помочь!

Доклады на заказ

Помимо общего списка направлений у нас есть и конкретные темы с уже готовыми описаниями и планами. Список докладов будет пополняться.

 [Смотреть список докладов](#)

Консультация с ПК

Если сомневаетесь в теме, то пообщайтесь с Программным комитетом напрямую на видеосозвоне или письменно.

 [Назначить митинг](#)

 [Написать](#)

ВО ВРЕМЯ РАССМОТРЕНИЯ

- ПИНГ

ВО ВРЕМЯ РАССМОТРЕНИЯ

- ПИНГ

- ДОП
КОНТЕНТ

ПУБЛИКА ХОДИТ В ТЕАТР СМОТРЕТЬ
ХОРОШЕЕ ИСПОЛНЕНИЕ ХОРОШИХ
ПЬЕС, А НЕ САМУ ПЬЕСУ: ПЬЕСУ
МОЖНО И ПРОЧЕСТЬ.

АЛЕКСАНДР НИКОЛАЕВИЧ ОСТРОВСКИЙ

СПАСИБО!

КУРАТОР
ДОКЛАДА



Семён Левенсон

📍 @sem_eleven

ВОПРОСЫ ПО
КОНФЕРЕНЦИИ



Дана Стоянова

📍 @danadst