



**17 октября 2026**

Екатеринбург,  
Хаятт Ридженси,  
800 участников

# **CALL FOR PAPERS 2026:**

**Какие доклады  
мы ищем**

от создателей CodeFest



**HardFest** — это конференция для инженеров, которые строят и эксплуатируют реальные системы. Будем говорить не про тренды и хайп, а архитектуру, код и решения. За всё о чём отвечаем в проде — масштабирование, отказы, ограничения железа и инфраструктуры, цену архитектурных компромиссов.

Только то, что реально работает или ломается, от эксперта к эксперту. Меньше докладов, больше практики, инженерных споров и общения.  
И при всей хардкорности.

Разработка

AI/ML

Автоматизация и эксплуатация

Железо

# РАЗРАБОТКА

*Трек про то, как проектируют, реализуют и масштабируют под нагрузкой сложные системы.*

## **ИНЖЕНЕРИЯ НЕДЕТЕРМИНИРОВАННЫХ СИСТЕМ**

- Как проектировать, тестировать и эксплуатировать систему, ядро которой вероятно (LLM или агент)?
- Проектирование, тестирование и эксплуатация вероятностных систем: контракты, eval-ы, наблюдаемость, границы доверия.

## **АРХИТЕКТУРА И СИСТЕМНОЕ ПРОЕКТИРОВАНИЕ**

- Архитектура высоконагруженных систем; DDD, event-driven, CQRS; распределённые системы.
- API design и интеграции; архитектурные паттерны и антипаттерны.
- Управление сложностью и цена решений.
- Рефакторинг легаси и «легаси-микросервисов».

## **СИСТЕМНАЯ АНАЛИТИКА И SDLC**

- Связка от требований к архитектуре, реализации и эксплуатации как единая инженерная модель.
- Моделирование предметной области и онтологии.
- Decision-first.
- Качество как инвариант («невозможность несоблюдения» вместо контроля).
- Человек и ИИ как со-разработчики, и что это делает с SDLC. Цикл смещается от написания кода к оркестрации и ревью того, что сгенерировала модель.

## АЛГОРИТМЫ И ПЕРФОРМАНС

- Perf-инженерия на грани железа.
- Latency/throughput-компромиссы.
- Язык машины.

## ИНСТРУМЕНТАРИЙ И ПРОЦЕССЫ

- Автоматизация разработки.
- Метрики качества кода и процессов.
- Ограничение хрупкости.

## ИНЖЕНЕРНАЯ ОТВЕТСТВЕННОСТЬ

- Про инженерную сторону ответственности: как принимаются решения, как удерживается суждение и как проводится техническая позиция — с реальными кейсами, ценой и компромиссами. Например: коммуникация в сложных технических конфликтах.
- Работа с когнитивной сложностью и информационной нагрузкой.
- Инженерное влияние без формальной власти.
- Принятие технических решений при неполной информации.
- Удержание инженерной строгости под давлением сроков.
- Передача технического контекста без потерь между командами и уровнями.
- Границы делегирования технических решений.

# AI/ML

*Трек про AI как инженерную систему в проде. Нужны ML/AI-инженеры, которые вывели модели в прод и отвечают за последствия: за нагрузку, стоимость, качество, инциденты.*

## **AI ENGINEERING & INFERENCE — ML В ПРОДЕ**

- Инференс под нагрузкой и его экономика: latency/throughput, батчинг, KV-cache, квантование, дистилляция, утилизация GPU.
- Свои модели против GPT/Claude и т.д. — с расчётом, где что выгоднее.
- LLM-приложения: RAG, агенты, оркестрация инструментов, работа с контекстом и его ценой.
- Инженерия недетерминированных систем: контракты, eval'ы, наблюдаемость LLM, гардрейлы.
- Безопасность и контроль моделей.

## **MLOPS & DATA INFRASTRUCTURE — ДАННЫЕ КАК ФУНДАМЕНТ**

- Lakehouse, стриминг, data pipelines, фичесторы, векторные/семантические слои, data contracts.
- Жизненный цикл и дообучение моделей: версионирование, воспроизводимость, дрейф, авто-ретрейн.
- Инфраструктура инференса на масштабе: автоскейлинг GPU, мультитенантность, шедулинг, «своё железо vs облако».

**Примеры тем докладов:**

- Как снизили стоимость инференса LLM в  $N$  раз: квантование, батчинг и что сломалось.
- Своя модель против GPT/Claude: где посчитали и вернулись обратно (и наоборот).
- RAG на  $N$  млн документов: где деградирует качество и сколько стоит контекст.
- Агенты в проде: контракты, eval'ы, наблюдаемость вероятностной системы.
- Постмортем: как галлюцинации LLM положили бизнес-метрику и что поменяли.
- Feature store под нагрузкой: архитектура, цена и грабли.
- Дрейф данных и авто-ретрейн: как ловим деградацию раньше пользователей.
- Экономика GPU-кластера: автоскейлинг, мультитенантность, FinOps инференса.
- Экспертная система + LLM: где логический ИИ спасает там, где статистика не тянет.

# АВТОМАТИЗАЦИЯ И ЭКСПЛУАТАЦИЯ

*Трек про жизнь системы в проде: как её собирают и катят без ручного труда, как держат под нагрузкой, наблюдают, защищают и чинят, когда она падает — и во сколько это обходится.*

*Нужны инженеры (Senior/Lead/Principal, SRE, platform-, DevOps- и security-инженеры, техлиды, СТО), которые отвечают за инфраструктуру и надёжность целиком и готовы честно разобрать не причёсанную success-story, а реальные компромиссы, цену решений и провалы.*

## **АВТОМАТИЗАЦИЯ — КАК ПЕРЕСТАТЬ ДЕЛАТЬ ЭТО РУКАМИ**

- Platform engineering и внутренние платформы — DevEx как продукт, self-service инфраструктура, разворачивание окружения «по кнопке».
- IaC и GitOps на масштабе — Terraform/Pulumi и т.д., ArgoCD/Flux и т.д.; не «мы перешли и довольны», а где ломается, дрейф стейта, цена перехода.
- CI/CD и delivery как инженерная система — прогрессивные и автоматические деплои (canary, blue-green), автоматические откаты и т.д.
- Auto-remediation и самовосстановление — автоматический ответ на отказ, автоскейл, автозаккрытие типовых инцидентов.
- Policy-as-code и автоматизация комплаенса/безопасности — OPA, автоматические проверки в пайплайне, хранение и ротация секретов.

- AI/агенты в эксплуатации — автотриаж инцидентов, разбор алертов с инженерной механикой и ценой ошибки, помощь в деплое. Как правильно ограничить работу агентов в проде и роль человека в обеспечении их работы.

## **ЭКСПЛУАТАЦИЯ — КАК СИСТЕМА ЖИВЁТ И ПАДАЕТ ПОД РЕАЛЬНОЙ НАГРУЗКОЙ**

- Kubernetes и cloud-native под ростом — где и с какими столкнулись ограничениями, что стоило по деньгам, как перепроектировали.
- Сети и трафик — L2/L3, балансировка нагрузки, CDN, проксирование, шлюзы (API gateway, ingress); где сеть становится узким местом и во сколько обходится.
- Экономика вычислений и FinOps — стоимость облака и его оценка/прогнозы, инференса и железа как проектное ограничение; где резали кости и что при этом сломалось.
- Observability на масштабе — трейсинг, SLO (автовыставление и отслеживание), наблюдаемость распределённых и недетерминированных (LLM/агентных) систем, борьба со стоимостью телеметрии.
- Постмортемы и реальные инциденты — что сломалось, сколько стоило, что поменяли в инженерной культуре, кросс-анализ инцидентов.
- Устойчивость — chaos engineering, нагрузочное и стресс-тестирование, incident management (дежурства, эскалации, платформы), «цифровой двойник» пользователей, не бумажные учения (отключения сервисов, ДЦ и т.д.).
- Security-by-design — безопасный по построению код, симуляция атак, риски цепочки поставок и опенсорса, какие инструменты security встроены в процесс и как, их влияние.
- Безопасность в эпоху агентов — новые классы угроз с появлением AI-агентов в проде: контроль их действий и прав, prompt injection, аудит и границы доверия к автономным системам.

# ЖЕЛЕЗО

*Трек про то, где софт упирается в физический мир и им же управляется. Инженерная механика на стыке hardware + software + data: честные компромиссы, цена решений, реальный масштаб и отказы. Не приветствуются обзоры девайсов.*

## УСТРОЙСТВА И СВЯЗКА HW + SW + DATA

- Проектирование и вывод в серию собственных устройств (хабы, станции, датчики, камеры, edge-железо, медтехника и прочее).
- Проектирование для различных сред эксплуатации (температура, влажность, конденсат, давление, вакуум, вибрации, агрессивные среды — спец. газы, соляной туман и т.д.).
- Связка прошивки, драйверов и облака в единую систему. Проблемы совместимости разных устройств на уровне прошивки, ревизий устройства; протоколы и механики обновления, чтобы не получить кирпич; вопросы защиты от несанкционированного доступа.
- Работа с требованиями заказчика, когда сам заказчик не знает, чего хочет (знает, но не может сказать).
- Сложности и подводные камни при сквозной разработке изделий.
- Примеры: «Путь устройства от прототипа до серии: где нас подвели тепло и питание»; «Edge-инференс на устройстве: латентность, память, батарея»; «OTA-обновления парка устройств в поле без кирпичей».

## РОБОТЫ И PHYSICAL AI

- Автономное передвижение на полигонах и в городе.
- Манипуляция в лаборатории и в реальности — на складе.
- Вопросы тестирования на основе модели (Software In the Loop) или на основе реального устройства (Hardware In the Loop).
- Восприятие пространства, сенсоры, принятие решений в движении.
- Real-time контуры под жёсткими ограничениями по latency и энергии; перенос стека на разные платформы.
- Взаимодействие между несколькими автономными юнитами.
- Примеры: «Складская робототехника: комплектация и сортировка под нагрузкой»; «Physical AI в городе: распознавание сцены и решения на скорости»; «Sensor fusion, когда датчики врут».

## МИКРОЭЛЕКТРОНИКА

- Сложности в проектировании и проверке самих микроконтроллеров и процессоров. Вопросы применения остальных компонентов.
- Борьба за такты и микроватты, тепловыделение, отладка на осциллографе.
- Компромиссы между стоимостью BOM, надёжностью и производительностью.
- Поддержка зоопарка прошивок и ревизий устройств, устаревших компонентов.
- Примеры: «Как мы ужали алгоритм в микроконтроллер за N килобайт»; «Perf-инженерия на уровне регистров и шин»; «Дешевле, надёжнее, быстрее — выбери два: реальный trade-off BOM».

## БЕЗОПАСНОСТЬ НА УРОВНЕ ЖЕЛЕЗА

- Защищённое хранение и обработка данных в устройствах.
- Защита от реверс-инжиниринга и лучшие практики реверс-инжиниринга готовых изделий.
- Безопасность автономного транспорта и роботов.
- Примеры: «Secure boot и защита ключей на массовом устройстве»; «Модель угроз для автономного транспорта».

## ПОСТМОРТЕМЫ АППАРАТНЫХ ОТКАЗОВ

- Что сломалось в поле, как чинили, что изменили в процессе разработки.
- Проблемы, с которыми пришлось мириться из-за очень дорогого фикса или по другим причинам.
- Примеры: «Партия устройств вышла из строя: разбор физической причины и цены»; «Отзыв прошивки: как мы ловили баг, который проявлялся только на морозе».

## ЭКСПЛУАТАЦИЯ И ЭКОНОМИКА «ЖЕЛЕЗНОЙ» ИНФРАСТРУКТУРЫ

- Защита от неправильного применения.
- Трассировка состояния устройств от монтажа до снятия с эксплуатации.
- Эксплуатация в различных средах.

## **ФОРМАТЫ:**

*помимо докладов, приветствуются интерактивные — инженерные демо-зоны и DIY-зона с живыми стендами.*



**ПОДАТЬ ЗАЯВКУ  
НА ДОКЛАД:**

<https://cfp.hardfest.ru/>

**ПРОГРАММНЫЙ КОМИТЕТ:**

[program@codefest.ru](mailto:program@codefest.ru)