

Наименование задачи (проекта)

Инженерное управление качеством решений, которые неинженерные роли создают с помощью ИИ-агентов: защитные ограничители и рабочие процессы

Описание проблемы

Каждый сотрудник Freedom Афиша получил ИИ-агента для написания кода с подпиской. Неинженеры (менеджеры продуктов, маркетинг, продажи, поддержка) теперь сами собирают решения, которых раньше месяцами ждали в инженерной очереди и на которые всегда не хватало ресурса.

Плюс – скорость и инициатива. Минус – скрытый технический долг: у неинженеров не хватает знаний проектировать решения безопасно, по инженерным стандартам и так, чтобы их можно было сопровождать. В компании накапливается:

- сервисы без ответственного, без тестов, без документации и без контроля состояния;
- риски безопасности: ключи и пароли прямо в хранилищах кода, уязвимые сторонние библиотеки, бесконтрольный доступ к данным, код без проверки;
- незапланированная нагрузка на инженеров, когда такие сервисы ломаются;
- отраслевой риск: эксперименты, видимые клиентам, затрагивают персональные данные и платёжный контур.

Простые ответы не работают: запрет уводит разработку в тень, сплошная проверка силами инженеров не масштабируется (десятки создателей на единицы инженеров), бездействие копит риск. Нужен средний путь: встроить стандарты не в людей, а в среду – в шаблоны, в автоматические проверки, в платформу, в правила. Ограничители незаметны, когда всё идёт штатно, и строги там, где риск реален.

Цель и описание задачи (проекта)

Спроектировать общекорпоративную модель, при которой неинженеры продолжают создавать решения с помощью ИИ-агентов (Claude Code, Codex, OpenCode), а результаты соответствуют стандартам качества, безопасности и сопровождаемости при минимальном участии инженеров.

Что спроектировать:

1. **Классификация по риску** – правила отнесения по измеримым признакам (чувствительность данных, доступность извне, уровень привилегий, влияние на бизнес), кто и когда присваивает класс. Для каждого класса свой режим ограничителей и свой уровень участия инженеров. Граничный случай: внутренний инструмент с правом записи в платежи.
2. **Целевые рабочие процессы** – путь решения от идеи до вывода в эксплуатацию и снятия с неё; где неинженер идёт один, а где подключается инженер.
3. **Конкретные ограничители** – механизмы, а не лозунги: эталонные шаблоны проектов; автоматические проверки в конвейере сборки (анализ стиля кода, тесты, статический анализ безопасности, поиск ключей и паролей, проверка сторонних библиотек, правила в виде кода); политики хранилищ кода (защита основных веток, обязательная проверка изменений назначенными ответственными, изоляция сред); платформенные границы (белый список разрешённых сервисов, песочница, управляемый доступ к данным); встроенный сбор метрик и оповещения. Для каждого ограничителя – где он срабатывает и что происходит при срабатывании.
4. **Модель ответственности** – кто отвечает за сервис, роли, правила передачи вопроса инженеру, судьба брошенных сервисов без хозяина.
5. **План распространения** – на всю компанию с низкой стоимостью тиражирования: самообслуживание и шаблоны, а не ручное сопровождение каждого создателя.
6. **Мотивация** – ограничители разрешают, а не блокируют; трение соразмерно риску; быстрый легальный путь вместо обхода правил.
7. **Нагрузка на инженеров и метрики** – упрощённая оценка трудозатрат инженеров (первоначальные вложения в платформу, регулярные проверки, исключения, разбор сбоев) с явными допущениями; метрики успеха с определениями; компромиссы – чем модель жертвует.

Рамки и вводные

- Решение не привязано к конкретному агенту: поддерживаются любые (Claude Code, Codex, OpenCode), без зависимости от поставщика.
- Опирается на уже имеющиеся систему контроля версий, конвейер сборки и доставки, средства сбора метрик и контроля состояния, а не заменяет их.
- Неинженеры создают что угодно: веб- и мобильные приложения, внутренние инструменты, доработки существующих продуктов.
- Отрасль Freedom Афиша – продажа билетов через интернет: персональные данные и платёжный контур.
- Масштаб качественный: десятки создателей на единицы инженеров. Точные цифры не задаются.
- Возможности имеющихся инструментов и допустимый доступ к персональным данным и платёжному контуру не детализированы: участник задаёт минимальный базовый уровень или явно фиксирует свои допущения.

Формат сдачи

Только проектный документ (в формате Markdown или PDF). Код и прототипы не требуются. Схемы – по желанию.

Критерии оценки (для жюри)

Обязательные условия (без них работа не засчитывается): решение не привязано к конкретному агенту и опирается на уже имеющиеся инструменты; есть классификация по риску с явными правилами отнесения; формат – проектный документ.

Оцениваемые качества (0 – нет, 1 – частично, 2 – полно):

- ☐ Строгость контроля зависит от риска, класс присваивается по измеримым признакам.
- ☐ Ограничители – конкретные механизмы с указанием места и поведения при срабатывании (отдельно от вопросов безопасности ниже).
- ☐ Участие инженеров сведено к минимуму и подтверждено оценкой трудозатрат с допущениями.
- ☐ Низкая стоимость распространения (самообслуживание, шаблоны).
- ☐ Мотивация сохранена: быстрый легальный путь, трение соразмерно риску.

- ☐ Модель ответственности полна: кто отвечает за сервис, передача вопроса инженеру, брошенные сервисы, вывод из эксплуатации.
- ☐ Безопасность: ключи и пароли, сторонние библиотеки, доступ к данным (персональные данные, платежи), риски цепочки зависимостей.
- ☐ Метрики с определениями и способом сбора; честные компромиссы.