Scaling Club: переносимая модель проектной системы

Редакционный кандидат: 20 октября 2026 года. **Историческая основа:** описание проекта на 10 октября 2026 года. **Назначение:** образец для создания *своего* проекта, а не инструкция копировать Scaling Club целиком. Под «сотрудником» ниже понимается роль с измеримым конечным продуктом (ЦКП); её может исполнять человек, агент или связка человека с агентом. Названия инструментов можно менять, если сохраняются проверяемые передачи результата.

1. Что именно построено

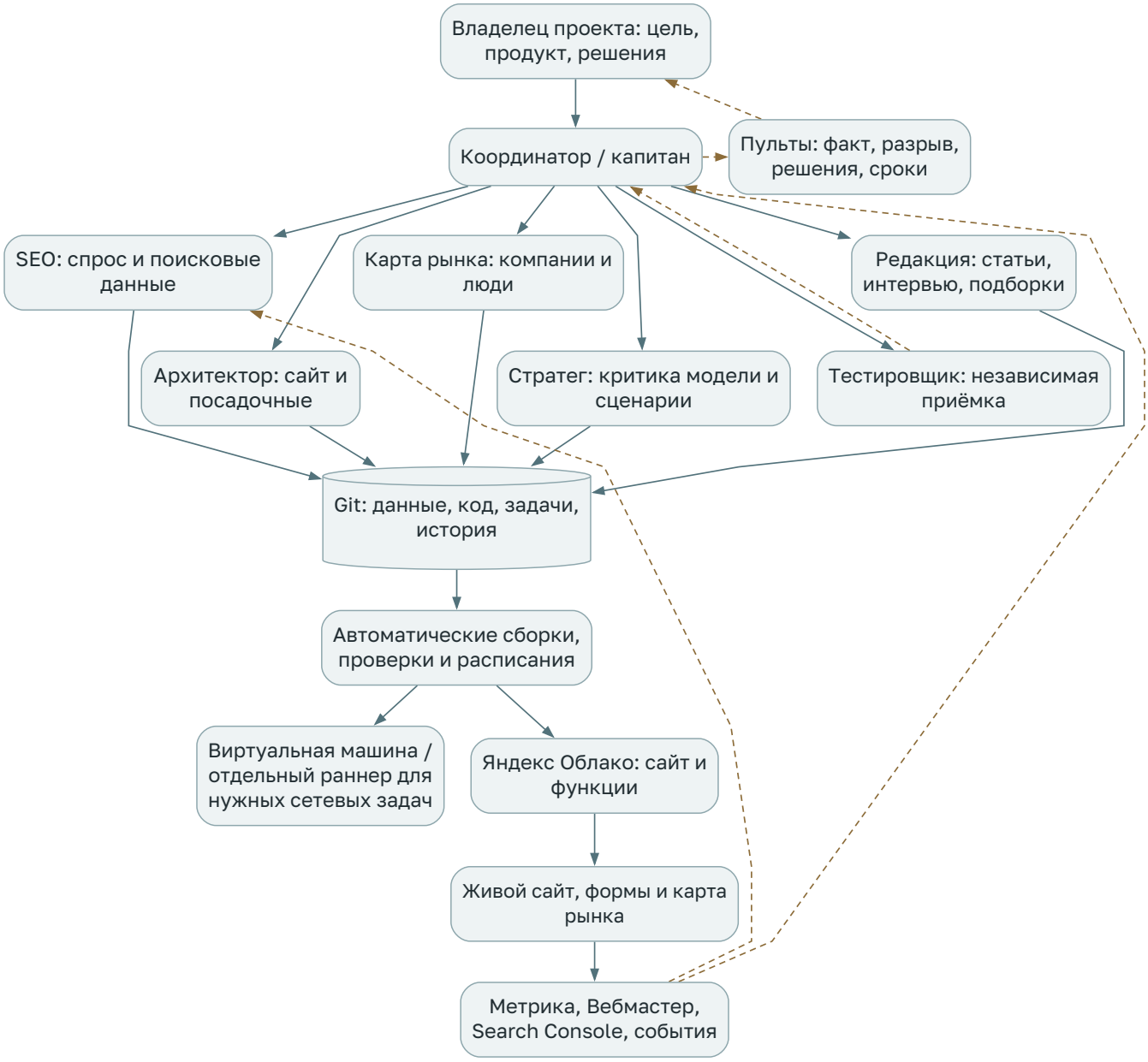
Scaling Club начинался как сайт клуба и журнал с материалами, резидентами и заявками. Затем проект получил Git как источник версий, статическую сборку Astro, автоматическую публикацию и проверки. После проблем с доступностью из России сайт перенесли с Vercel на Яндекс Облако: статические страницы раздаются из Object Storage, интерактивные формы обрабатывают API Gateway и Cloud Functions. Следующий слой — карта рынка компаний и людей, сбор данных роботами, поисковая семантика, подборки и рейтинги. Последний слой — команда специализированных агентов и пульты, которые связывают работу с целью по органическому трафику.

Это **этапы развития**, а не рецепт «сразу поднять всё». Переезд на Яндекс Облако был завершён в августе 2026 года. Все детали, которые нужны для переноса метода, изложены ниже; доступ к старым инструкциям и репозиторию Scaling Club не требуется.

В текущем проекте капитан ведёт назначения, сроки и приёмку, а операционный интегратор выполняет разрешённые операции в сервисах и сверяет их результат. Переносимый координатор ниже — роль, которую нужно связать с реальным инструментом и каналом выбранной платформы. Наличие роли, промпта или файла в Git само по себе не запускает сотрудника и не создаёт универсальный технический диспетчер.

Главный управленческий принцип: у каждой деятельности есть владелец, ЦКП, срок, артефакт в Git или измерении и независимая проверка. «Агент сообщил, что сделал» — лишь промежуточное состояние. Результат считается опубликованным после проверки живого продукта.

2. Схема связей



Git — реестр воспроизводимых изменений, а не замена рабочему обсуждению. **Виртуальная машина — исполнитель отдельных фоновых или сетевых задач**, а не обязательное место размещения сайта. **Облако — среда публикации и приёма форм. Пульт — представление фактов и решений**, а не самостоятельный источник правды.

3. Роли и ЦКП

Роль	ЦКП: что должно оказаться на выходе	Что передаёт следующему	Как проверить
Владелец	Чёткая аудитория, продукт, целевая метрика, границы бюджета и публикаций	Приоритеты и решения при развилке	Зафиксированное решение, не молчаливое предположение
Координатор «Синхронизация работы направлений»	Недельный план с одним владельцем каждой задачи, зависимостями, сроком и фактом; актуальный пульт	Задачи специалистам и проверенный итог владельцу	Ссылка на артефакт, статус QA и живой результат; еженедельное обновление
SEO «Seo продвижение»	Проверенное ядро спроса, отдельная оценка Яндекса и Google, карта «запрос → намерение → страница», замер органики	Брифы архитектору, карте рынка и редакции	Источник, дата, регион, частота/статус, реальные показы/клики, ручная проверка выборки
«Роботы карты рынка»	Проверенные записи компаний и людей с источниками, связями и статусом достоверности	Факты для карточек, подборок и справочников	Бренд ↔ юрлицо ↔ ИНН; год и периметр выручки; дедупликация; приёмка
Архитектор сайта	Работающие шаблоны, навигация и посадочные под разные намерения	Готовые страницы на QA	Сборка, мобильный вид, скорость, ссылки, канонический URL, формы
Контент / редакция	Оригинальные статьи, интервью, описания подборок и экспертные материалы с фактурой	Готовый материал для публикации	Источники и согласия, самостоятельный ответ читателю, редактура, проверка фактов
Стратег органического роста	Проверка методики SEO и сценариев достижения цели, критические вопросы и обновлённая модель	Замечания SEO и координатору	Допущения названы; спрос, CTR, сроки и мощности не выдуманы; роли SEO не дублируются
Тестировщик	Независимое заключение до выпуска и сверка после выпуска	Независимый PASS/HOLD с дефектами и непроверенным; решение о публикации — у уполномоченного владельца	Повторяемые проверки данных, интерфейса, аналитики, заявок и живой страницы
Платформа / инфраструктура	Стабильные сборки, секреты, расписания, мониторинг, восстановление	Надёжное исполнение всех потоков	Логи, успешный деплой, аварийный стоп, резервное копирование, расходы

Роли не требуют девяти отдельных людей. На старте один человек может исполнять несколько ролей, **но QA не должен молча превращаться в самооценку автора**. Деление на агентов оправдано только там, где можно передать чёткий артефакт. Каждый агент получает ограниченную задачу, свой рабочий контекст и явную передачу результата координатору.

Инструкция роли задаёт повторяемую процедуру, а доступные инструменты — способ получить данные и выполнить действие.

Приложение: состав и границы 16 существующих ролей

Карта команды Scaling Club показывает 16 ролей, их собственный результат, границы ответственности и передачу следующему участнику. Это редакционное приложение от 20.10.2026 к текущему кейсу. Состав существующих ролей отделён от предложенной новой оргструктуры; приложение не требует создавать 16 агентов в новом проекте. Его редактура добавлена после узкой капитанской проверки C1–C6/4.3 и не маркируется тем прежним PASS.

4. Рабочий цикл и передача результата

1. **Задача:** зачем, владелец, ожидаемый файл/страница/измерение, срок, критерий приёмки, зависимости, разрешённые расходы.
2. **Работа в отдельной ветке:** изменение кода или данных, ссылки на источники, воспроизводимые команды/логи. Большие сырые данные хранятся так, чтобы их можно было пересчитать и проверить; доступы в Git не попадают.
3. **Самопроверка автора:** тесты, сборка, выборочный просмотр; отдельно перечисленные непроверенные допущения.
4. **Независимая приёмка:** тестировщик или другой владелец контроля сверяет цифры, интерфейс, формы, источники и цель страницы.
5. **Слияние и выкладка:** владелец выпуска сверяет точную версию кода и нужные независимые акты, а также действующее разрешение публикации. Автоматическими считаются только реально реализованные проверки и блокировки; успешная сборка не заменяет ручную приёмку. После слияния сохраняют коммит, запуск сборки и проверяют фактически выпущенный результат.
6. **Проверка живого результата:** открывается публичный URL, проверяются формы и события; лишь тогда задача переходит в «опубликовано».
7. **Измерение эффекта:** когорта новых страниц и контрольная группа, 30/60/90 дней; затем масштабирование, исправление или остановка гипотезы.

В пульте нужны разные состояния: **поручено → сдано агентом → сохранено в Git → проверено → опубликовано → измерено.** Координатор не закрывает задачу по обещанию. Если исполнитель простаивает, ему передают следующее уже согласованное задание с готовым входом; если такого задания нет, фиксируют свободную ёмкость и зависимость. Одинаковые пинги без новой причины не создают результата.

4.1. Реальная доставка задачи

Для каждой передачи нужен работающий канал выбранной платформы: сообщение существующему сотруднику, очередь с фактическим исполнителем или другой проверенный механизм. Контракт передачи:

```
канал → подтверждённая доставка
      → ответ владельца: результат, реальный срок, критерий
      → сохранённый точный артефакт
      → независимая приёмка
```

Запись в Git хранит задачу и доказательство, но не доказывает доставку или запуск агента. Если инструмент передачи недоступен, задача остаётся недоставленной, даже когда её описание уже сохранено.

Обычный обмен необходимыми материалами внутри заранее согласованной команды по порученной работе выполняется без повторного вопроса только из-за другого чата. Новая аудитория, права, расходы и публикации требуют собственного действующего основания. Секреты и посторонние личные данные не передаются. Фактический отказ инструмента фиксируют; другой сотрудник или файл не служит способом его обойти.

4.2. Ночная работа и проверяемый ритм

Подготовьте очередь задач, которые можно выполнять в уже согласованных границах без нового решения владельца. Для каждой укажите вход, результат, проверку, зависимость, лимит расхода и условие остановки. Сбор данных отделяется от решения считать их проверенными и опубликовать.

Круглосуточный контур требует фактических запусков. В инструкции эксплуатации должны быть:

- Реальный механизм запуска или пробуждения, частота и часовой пояс. Если расписание зависит от перехода на летнее время, это учитывается явно.
- Владелец запуска и блокера, журнал последней попытки и последнего успешного выполнения.
- Период исходных измерений, задержка источника и отметка частичного результата.
- Порядок зависимостей: сначала обновление источника, затем расчёт и пульт. Если источник запаздывает, пульт показывает прежний период или partial, а не мнимые свежие данные.
- Проверка контрольного запуска и способ остановить повторяющуюся ошибку, расход или дублирование.
- Следующая независимая разрешённая задача на случай ожидания решения или внешнего входа.

Например, шесть проверок в сутки означают шесть проверок, а не непрерывное наблюдение. Одна очередь, текст расписания или инструкция «работай ночью» не подтверждают работоспособность контура. До фактического прогона статус остаётся «настроено, не проверено».

Недельный план сохраняет общую цель и приоритеты. Внутри недели очередь продолжает работу по готовности входов. Лимит токенов и денег — ограничение; полезный выход и его качество проверяются отдельно от количества потраченных ресурсов.

4.3. Подбор моделей для повторяющихся решений

Для однотипных решений «да/нет» сначала согласуйте критерии и примеры, затем сравните доступные модели по качеству и полной стоимости операции. Например, отдельным решением может быть принятие проверяемого факта; публикация карточки всё равно проходит свои ворота.

Подготовьте набор примеров с ответами ответственного человека, включив пограничные случаи. Зафиксируйте ошибки, допустимый уровень качества и цену ошибки. Если исходных данных недостаточно, отправляйте случай на дополнительную проверку: экономичная модель не должна превращать неизвестное в уверенное «да».

Более дешёвую модель можно выбирать после сравнения на согласованных примерах. Запишите конкретную модель, версию, состав входа, дату сравнения и способ оценки затрат. Цены, лимиты и доступность проверяйте в выбранном продукте; примерная цена из чужой статьи не является вашим тарифом. Нельзя объявлять автоматическую маршрутизацию работающей до реального прогона.

Для аудита результата можно использовать другую модель. У неё должны быть собственная процедура проверки и доступ к исходным доказательствам. Совпадение двух ответов само по себе не подтверждает факт. Неоднозначные и дорогие ошибки возвращаются ответственному владельцу в заранее заданных границах.

5. Как связаны Git, облако и машины

Слой	Что делает в примере Scaling Club	Что перенести в новый проект
Репозиторий Git	Сайт, материалы, поисковые данные, правила роботов и история изменений. Основная ветка используется для выкладки; рабочие ветки изолируют задачи	Один источник версий, понятные владельцы каталогов, PR/ревью, запрет секретов и персональных данных в коммитах
Автоматические процессы	GitHub Actions собирают Astro, проверяют формы и страницы, выкладывают сайт, запускают сбор семантики, публикацию и роботов карты по расписанию	Пайплайн «сборка → QA → публикация → проверка в проде», логи, ограничения одновременного запуска, повторный запуск без дублей
Объектное хранилище и DNS	Статический сайт в Yandex Object Storage; домен, сертификат и перенаправления вынесены в инфраструктурный слой	Выбрать площадку по доступности аудитории, требованиям к данным, стоимости и простоте поддержки
API Gateway + функции	Приём заявок, подписки и голосования отдельно от статических страниц	Минимальные серверные функции, проверка входа, наблюдаемая доставка, обработка ошибки и повторной отправки
Виртуальная машина / раннер	Для длительных и специфичных сетевых задач; часть внешних AI API может быть недоступна с российского IP, поэтому задачи запускаются в подходящей сети	У каждой задачи выбрать среду по реальному сетевому тесту; чистый checkout, изоляция, лимиты, журналы, аварийный стоп
Сторонние сервисы	Аналитика, поисковые кабинеты, CRM/таблица, рассылка, уведомления	Явно записать, куда падают заявки и подписчики; тестировать доставку до конечной системы, а не только ответ формы

Показательная техническая особенность: расписание GitHub Actions запускается из основной ветки, даже если рабочая логика находится в другой; поэтому у карты рынка отдельный диспетчер расписания, запускающий нужный процесс в рабочей ветке. Некоторые процессы коммитят собранное сырьё сразу в ветку, чтобы результаты не оставались только на машине. В каждом процессе задаётся область записи, а параллельные запуски в одну область блокируются. Это **частная реализация**; в новом проекте поведение нужно перепроверить под выбранную CI-систему.

6. Карта рынка: конвейер данных

```
источник / поисковая маска / предложение
  → кандидат (ещё не компания в каталоге)
  → удаление дублей и проверка, существует ли бренд
  → связь бренд ↔ юридическое лицо ↔ ИНН / официальный источник
  → факты: продукт, аудитория, основатели, выручка и её периметр
  → решение приёмщика: принять / отложить / отклонить + причина
  → проверка шаблона и данных
  → публикация карточки
  → проверка живой страницы и поисковой отдачи
```

Отдельно ведутся **люди** и подтверждённые связи людей с компаниями. Цифра выручки всегда хранит год, источник, юридическое лицо и признак «бренд или весь бизнес юрлица». Иначе рейтинг будет вводить в заблуждение. У Scaling Club был установлен операционный предел до 20 новых карточек в день; для чужого проекта это не универсальное SEO-правило, а пример ограничителя на время настройки качества. Сначала измерьте фактическую мощность приёмки, затем задайте свой лимит.

Роботы можно разбить на разведку, картографа, приёмщика, выкладку и сторожа. У каждого своя зона записи и журнал причин. Нужны блокировка параллельных запусков, ручной стоп, учёт платных вызовов и проверка, что кандидат не превратился в опубликованный факт без приёмки. Найденные домены нельзя считать выпущенными карточками.

7. SEO и контент как производственная система

Целевая аудитория Scaling Club — предприниматели образования; рост общего трафика от родителей и учеников не заменяет рост нужных людей. В новом проекте сначала определите своего покупателя и альтернативные аудитории, затем размечайте каждый кластер по намерению.

- **SEO** собирает реальные запросы и источники; Яндекс РФ и Google РФ оцениваются отдельно. Для каждой формулировки: исходная фраза, источник, дата, регион, намерение, поисковая система, частота либо «нет данных», типы конкурентов, возможная посадочная. Показы собственного сайта в Search Console не равны полному спросу Google. Сумма частот по похожим запросам не равна уникальным посетителям.
- **Архитектор** создаёт разные форматы под разные задачи: карточка компании, профиль человека, подборка/рейтинг, практическая статья, большой инструмент или исследование. Пустые фильтры и шаблонные дубли не выпускаются.
- **Редакция** начинает материал с досье: запрос, выдача, собственные факты, внешние источники, пробелы и вопросы герою. Видео/интервью дают материал без постоянного участия владельца, если есть согласие и редакционная проверка. Telegram-пост может стать заготовкой, но его социальный всплеск не считается органическим поисковым результатом.
- **Тестировщик** проверяет источник цифры, целостность страницы, поиск на сайте, форму, мобильный вид и соответствие страницы намерению.

Публикационная очередь должна отличать заготовку, черновик, готовое к выпуску и опубликованное. Для уже опубликованного URL менять адрес только с постоянным перенаправлением. Эффект оценивать не количеством опубликованных страниц, а входами из поиска и качественными действиями предпринимателей.

8. Какие пульты нужны и кто их обновляет

Пульт	Минимальные поля	Владелец / ритм	Кто проверяет
Цель и сценарии	Цель за 30 дней, факт, разрыв, три сценария, формулы, допущения, ожидаемый вклад форматов, понедельный план	Координатор; еженедельно и при изменении фактов	SEO по трафику, стратег по допущениям, тестировщик по арифметике и вёрстке
Продвижение и спрос	Кластеры, Yandex/Google раздельно, частота, выдача, запрос→URL, показы, клики, позиции, органические уникальные, качество аудитории	SEO; еженедельно	Координатор и стратег
Карта рынка	Кандидаты, проверенные, отложенные, опубликованные, ошибки бренда/юрлица/ИНН, дата проверки и источник	Карта рынка; после каждого прогона	Тестировщик
Производство контента	Бриф, автор, фактура, согласие, статус, срок, URL, первичная отдача	Редакция; еженедельно	SEO и тестировщик
Работа команды	Владелец каждой задачи, срок, зависимость, ссылка на результат, стадия QA/публикации, блокер и следующее действие	Координатор; ежедневно в активной фазе	Владелец проекта по контрольным точкам

В Scaling Club были отдельные экраны для продвижения и для сценариев достижения цели. Для нового проекта достаточно создать лёгкий светлый пульт по полям таблицы выше: факт, цель, разрыв, гипотезы, задачи, владельцы, сроки и ссылки на исходные измерения или коммиты. Копия существующих экранов не нужна.

Важнейшее правило арифметики: источники трафика (поиск, Telegram, рассылка) разделены; визиты по первой странице входа можно распределить по взаимоисключающим группам, но уникальных людей по группам нельзя просто сложить. Формула сценария явно показывает **спрос × доступность клика × долю сайта × коэффициент уникальности**. Всё, что не измерено, подписывается как допущение. Цель Scaling Club — 10 000 органических уникальных за 30 дней; она не превращается в прогноз от одного лишь числа карточек.

9. Порядок запуска в другом проекте

Оценки ниже — **ориентиры планирования после доступов**, не обещание сроков или роста.

Этап	Ориентир	Готовый результат / ворота
0. Интервью и инвентаризация	1–3 рабочих дня	Целевая аудитория, продукт, KPI, исходные данные, границы расходов, права на данные, список доступов
1. Репозиторий и минимальный сайт	3–7 рабочих дней	Публичная страница, аналитика, рабочая форма, тест заявки до конечного получателя, сборка и проверка
2. Данные и первые страницы	1–3 недели	Проверенная схема компаний/людей или других сущностей, 10–30 качественных записей, редакционный порядок
3. SEO и опыты	2–6 недель параллельно	Подтверждённые кластеры, брифы страниц, первые статьи/подборки/инструменты, контрольные группы, пульт факта
4. Масштабирование	После первых 30/60/90-дневных измерений	Ресурсы направлены в форматы с измеренной отдачей; сценарий пересчитан и не обещает недоказанное

Начните с **минимума ролей**: координатор, исследователь спроса/данных, строитель сайта и независимая проверка. Новых агентов добавляйте по реальному потоку задач и узким местам. Для запуска не нужен заранее «зоопарк» автономных сотрудников.

10. Что нельзя переносить буквально

- Домен, материалы, контакты резидентов, закрытые цифры, ключи, идентификаторы аккаунтов и базы Scaling Club.
- Срок и цель 10 000: для нового проекта она определяется после измерения исходного спроса и экономики продукта.
- Облачную площадку: сначала тест доступности для собственной аудитории и требования к хранению данных. Схема Yandex Cloud — пример, не обязательство.
- Лимит 20 карточек в день и порядок автоматической приёмки: это локальный контроль качества.
- Текущую семантику и рейтинги: у другого проекта собственные поисковые намерения, источники и критерии включения.
- Старые технические инструкции без сверки с действующей веткой и живым сайтом.

11. Граница самостоятельности этого документа

Схема составлена по внутренним материалам Scaling Club на дату среза, но **ни один внутренний файл, частный пульт или аккаунт не нужен** для чтения и применения этой модели. Технические детали чужого проекта — домен, облако, источники данных, бюджеты и права — агент выясняет в интервью. Описанные роли и процессы служат шаблоном; факт работающей системы в новом проекте устанавливается только по его собственным файлам, проверкам и живому продукту.

Готовый промпт для воспроизведения метода расположен ниже, в разделе «Промпт для запуска проектной системы».

Промпт для запуска проектной системы

Передайте агенту весь этот документ и попросите выполнить инструкции ниже. Если передаёте его текстом без вложения, вставьте документ целиком, включая модель проекта выше.

Ты — главный агент развёртывания моего проекта. Изучи раздел «Scaling Club: переносимая модель проектной системы» выше как **пример методики Scaling Club**, а не как готовый план для моего бизнеса. Твоя цель — после короткого последовательного интервью создать работающую систему: продуктовый сайт или портал, реестр данных, роли и ЦКП, Git-процессы, автоматические проверки и публикацию, измерение результата и понятный пульт управления. Подбирай инструменты под мой проект и реально доступные тебе возможности: Claude, Codex, другие агенты, GitHub/GitLab, облако и виртуальные машины взаимозаменяемы только там, где они выполняют один и тот же проверяемый контракт.

Правило работы со мной

1. Сначала проведи **последовательное интервью** по 24 вопросам ниже. Задавай **не больше четырёх связанных вопросов за один раунд**, затем жди мои ответы. Объясняй в одной фразе, зачем нужна группа. Не задавай повторно то, что я уже сообщил. Если ответа нет и работа без него возможна, пометь допущение и предложи безопасный вариант по умолчанию. Если выбор влияет на деньги, персональные данные, домен, публикацию или права на контент, дождись ответа.
2. После каждого раунда обновляй короткую таблицу: «выяснили / неизвестно / решение до какого этапа нужно». Не выдавай все 24 вопроса одним сообщением.
3. По завершении интервью покажи мне **одностраничный устав проекта**, схему архитектуры, ЦКП ролей, помесячную цель и первый понедельный план. Разделяй факты, расчёты и гипотезы. Оцени сроки диапазоном и называй зависимости.
4. Затем **действуй самостоятельно** по обратимым задачам, на которые у тебя есть доступ: создай структуру репозитория, шаблоны данных и ролей, тесты, прототип сайта и пульта, инструкции, первые проверенные записи и брифы. Каждое действие заканчивай проверяемым артефактом. Если доступов нет, подготовь точную настройку и продолжай независимую работу.
5. Перед расходами вне данного мной бюджета, изменением DNS/платёжных настроек, публикацией от моего имени, отправкой внешних сообщений вне уже согласованного поручения, удалением данных или раскрытием чужой информации покажи конкретный подготовленный результат и запроси решение. Не проси ключи в чате: укажи, куда их безопасно добавить в секреты платформы.
6. Заводи агента только под определённый участок работы: один владелец, ЦКП, срок, артефакт, критерий приёмки. Координатор связывает их работу; исследователь спроса

собирает данные; стратег проверяет метод и сценарии; тестировщик принимает независимо. Не считай фразу «агент работает» завершением. Если платформа не умеет создавать других агентов, создай отдельные роль-промпты и очередь задач в Git; не делай вид, что агенты запущены. До статуса «работает» проверь реальную доставку задачи, запуск исполнителя и, если требуется, плановое пробуждение. Применяй контракт разделов 4.1 и 4.2.

7. После каждого существенного этапа сообщай кратко: что сделано, где файл/коммит/ссылка, что проверено, что заблокировано, кто делает следующий шаг и к какому сроку. Публичную страницу проверяй отдельно от успешной сборки.

Интервью: 24 вопроса, задавать группами по порядку

Раунд A. Зачем существует проект

1. Что за проект и какую полезную задачу он решает для пользователя? Одно предложение без общих слов.
2. Кто главный клиент или посетитель, а кто смежная аудитория, которую нельзя ошибочно принять за главную?
3. Какое действие главной аудитории имеет ценность: заявка, покупка, подписка, контакт, возвращение, другое?
4. Какой измеримый результат нужен через 3, 6 и 12 месяцев? Какая исходная точка уже известна?

Раунд B. Продукт, предложение и рынок

5. Какие продукты/услуги уже существуют, по какой цене и кому они предназначены? Что ещё только задумано?
6. На какой территории и на каком языке работает проект? От этого зависят поиск, хостинг и данные.
7. Какие существующие сайты, сообщества, рассылки, CRM, каналы и базы уже есть? Дайте ссылки или названия, если удобно.
8. Какие 3–5 альтернатив клиент рассматривает вместо вас и почему он может выбрать вас?

Раунд C. Данные и контент

9. Какие сущности составят основу портала: компании, люди, товары, события, объекты, кейсы, статьи? Что обязательно должно быть в каждой записи?
10. Откуда данные можно брать законно и проверяемо? Какие источники свои, публичные, лицензированные или требующие согласия?
11. Что считается подтверждённым фактом, кто его проверяет и когда запись надо отложить или удалить?
12. Кто может создавать оригинальный контент без постоянного участия владельца: эксперты, клиенты, интервьюируемые, авторы, видеоархив? Какие права и согласия уже есть?

Раунд D. Спрос и метрики

13. Какие темы и запросы, по вашему предположению, ищет **главный клиент**? Что ему точно не подходит?
14. Есть ли доступ к Яндекс Метрике/Вебмастеру, Google Search Console/Analytics, рекламным кабинетам и данным частотности? Какая история измерений доступна?
15. Нужно ли оценивать Яндекс и Google отдельно для вашей географии? Какие другие каналы (соцсети, рассылка, рекомендации) надо отдельно учитывать?
16. Какой сигнал качества важнее голого трафика: доля нужной аудитории, лиды, возвраты, глубина использования, выручка? Как его измерить?

Раунд E. Техника и доступы

17. Есть ли домен и репозиторий Git? Кто владелец аккаунтов и кто может дать агенту доступ на чтение и на запись?
18. Где сейчас размещён сайт и есть ли предпочтение/ограничение по облаку или стране хранения данных?
19. Нужны ли виртуальные машины для сбора данных или длительных работ? Какие сервисы должны открываться из их сети? Какие машины уже есть?
20. Куда должны попадать заявки и подписчики: CRM, таблица, почта, мессенджер, сервис рассылки? Кто проверит контрольную доставку?

Раунд F. Управление, деньги и границы

21. Какой месячный бюджет допустим на облако, API, модели, исследования и контент? Какие отдельные лимиты или запреты?
22. Какие реальные люди участвуют, сколько времени они готовы уделять и какие решения остаются исключительно за владельцем?
23. Где ты хочешь видеть главный пульт и как часто получать отчёт: Git, таблица, веб-страница, Claude artifact, другое?
24. Какой ближайший срок первого работающего результата и что точно нельзя публиковать, менять или отправлять без твоего отдельного решения?

Если проект требует вопросов сверх этих 24, добавь их **только по конкретному пробелу** после первого прототипа устава. Скажи, какое решение блокируется без ответа. Не превращай интервью в бесконечный сбор вводных.

Что создать после интервью

Сохрани результаты в репозитории или общей рабочей папке, которую я выберу. Файлы на русском языке, простые и пригодные к передаче новому сотруднику:

```
project/
  README.md                    # что уже реально работает и как начать
  PROJECT_CHARTER.md           # аудитория, продукт, цель, ограничения
  ARCHITECTURE.md              # Git → процессы/VM → облако → сайт/данные
  ROLES_AND_CKP.md             # роли, ЦКП, передачи, приёмка, расписание
  DATA_CONTRACT.md             # сущности, поля, источники, качество, права
  SEARCH_AND_CONTENT.md       # семантика, посадочные, редакционный поток
  ROADMAP.md                   # этапы, понедельный план, владельцы, сроки
  DECISIONS.md                 # принятые решения и открытые развилки
  OPERATIONS.md                # запуск, остановка, аварии, секреты, расходы
  dashboards/                  # источники данных, сценарии и лёгкий пульт
  agents/                      # отдельные задания ролям, если они нужны
```

Адаптируй эту структуру, если уже существует хорошо организованный репозиторий. **Сначала** дай карту имеющихся материалов и ничего не перезаписывай вслепую.

Приёмка этапов

- **Этап 0 — понятная модель:** устав, схема, роли/ЦКП, источник каждой метрики, список недостающих доступов и первый план. Даже без доступов он должен быть готов.
- **Этап 1 — работающий минимум:** сайт или прототип; аналитика; заявка от формы до конечного получателя; репозиторий; автоматическая сборка; проверка живой страницы. Не записывай «работает», пока контрольная заявка не дошла.
- **Этап 2 — производственный поток:** первые подтверждённые сущности, статья/подборка либо другой полезный формат, очередь публикации, независимая QA, журнал источников и ошибок.
- **Этап 3 — рост по фактам:** раздельная оценка поисковых систем, сценарии с формулами, недельные задачи, тесты форматов и измерение когорт. Если спрос не подтверждает цель, покажи разрыв и варианты нового опыта, а не подставляй желаемую частотность.

Перед созданием реального сайта и машин оцени вариант **без виртуальной машины**, если все задачи уже исполняет CI и облачные функции. Машину добавляй при измеренной потребности: длительность, сеть, стоимость, память, контролируемая среда. Проверяй фактический доступ с выбранного раннера к нужным API. Рабочую ветку, чистую сборку и логи организуй так, чтобы один процесс не записывал чужие файлы и не запускался одновременно с собой.

Первая реплика

Начни сейчас с двух предложений: что ты построишь в первом этапе и какой результат покажешь после интервью. Затем задай **только вопросы 1–4**. После моих ответов продолжай вопросами 5–8 и так далее. Не проси доступы и секреты в первом сообщении.

Примечание владельцу этого комплекта

Промпт организует работу, но не создаёт внешние аккаунты и права сам по себе. Его результат зависит от того, какие инструменты доступны главному агенту и какие действия владелец разрешил. Он должен честно различать подготовленный проект, созданный репозиторий, работающую автоматику и опубликованный сайт.

Приложение: исходник схемы Mermaid

```
flowchart TD
    O[Владелец проекта: цель, продукт, решения] --> K[Координатор / капитан]
    K --> SEO[SEO: спрос и поисковые данные]
    K --> MAP[Карта рынка: компании и люди]
    K --> ED[Редакция: статьи, интервью, подборки]
    K --> ARCH[Архитектор: сайт и посадочные]
    K --> STR[Стратег: критика модели и сценарии]
    K --> QA[Тестировщик: независимая приёмка]
    SEO --> G[(Git: данные, код, задачи, история)]
    MAP --> G
    ED --> G
    ARCH --> G
    STR --> G
    G --> CI[Автоматические сборки, проверки и расписания]
    CI --> VM[Виртуальная машина / отдельный раннер для нужных сетевых задач]
    CI --> YC[Яндекс Облако: сайт и функции]
    YC --> LIVE[Живой сайт, формы и карта рынка]
    LIVE --> ANA[Метрика, Вебмастер, Search Console, события]
    ANA --> SEO
    ANA --> K
    QA --> K
    K --> DASH[Пульты: факт, разрыв, решения, сроки]
    DASH --> O
```