

12+ Создание digital-продуктов с помощью ИИ в 2025 году: от идеи до релиза



Владимир Кафитулов

Владимир Кафитулов

**Создание digital-продуктов
с помощью ИИ в 2025 году:
от идеи до релиза**

«Издательские решения»

Кафитулов В.

Создание digital-продуктов с помощью ИИ в 2025 году: от идеи до релиза / В. Кафитулов — «Издательские решения»,

ISBN 978-5-00-671583-7

Искусственный интеллект меняет правила игры в создании digital-продуктов. Как перейти от хайпа к реальной пользе и создать продукт, который будет востребован? Эта книга — ваш практический навигатор в мире ИИ-разработки по состоянию на весну 2025 года. Независимо от того, новичок вы или опытный специалист, здесь вы найдете все необходимое, чтобы использовать мощь нейросетей на каждом этапе создания продукта — от первой идеи до поддержки и масштабирования.

ISBN 978-5-00-671583-7

© Кафитулов В.
© Издательские решения

Содержание

Введение	6
1.1. Почему весна 2025 – подходящее время для старта с ИИ	7
1.2. Наш подход: «человек + ИИ»	8
1.3. Как читать книгу	9
1.4. Этика и Ответственное использование ИИ	11
Часть I. Полный обзор ИИ-ландшафта на весну 2025	12
2.1. Запущенные модели	13
2.2. Российские решения	14
2.3. Анонсы ближайших релизов (Ожидания на конец 2025 – 2026) (С высокой степенью неопределенности)	15
2.4. Сравнительная матрица по ключевым задачам	16
2.5. Типы аккаунтов и тарифы	17
Часть II. Начало работы с ИИ	18
3.1. Веб- и мобильные интерфейсы (без использования API)	19
3.2. Интеграция через API: квоты и тарифы	21
3.3. Примеры использования API при разработке	22
3.4. Управление ключами и безопасностью API	23
Часть III. Prompt-инжиниринг и мульти-ИИ-воркфлоу	24
4.1. Базовые техники промпт-инжиниринга (Prompt Engineering)	25
4.2. Оркестрация нескольких моделей (Multi-AI Workflow)	26
4.3. Ловушки при мульти-ИИ-подходе (Common Pitfalls)	27
4.4. Практические примеры промптов для разработки	28
Часть IV. Код и автоматизация разработки	29
5.1. Генерация кода: от простых скриптов до сложных библиотек	30
5.2. Интеграция через API: примеры на Python и JavaScript с учетом российских платформ	32
5.3. Авто-тесты и regression suite: помощь ИИ с учетом российских моделей	35
5.4. CI/CD-пайплайн с шагами ИИ: приоритет российским ИИ для текстовых задач	36
Часть V. Дизайн, иллюстрации и мультимедиа	38
6.1. Создание визуального контента: сравнение Kandinsky, Midjourney, Stable Diffusion, DALL·E	39
Конец ознакомительного фрагмента.	40

Создание digital-продуктов с помощью ИИ в 2025 году: от идеи до релиза

Владимир Кафитулов

© Владимир Кафитулов, 2025

ISBN 978-5-0067-1583-7

Создано в интеллектуальной издательской системе Ridero

Введение

Добро пожаловать в «Создание digital-продуктов с помощью ИИ в 2025 году: от идеи до релиза» – руководство, основанное на опыте писателя, разработчика, маркетолога, продуктового менеджера и дизайнера. Наша цель – не просто рассказать о возможностях нейросетей, но и показать вам путь через весь процесс разработки продукта: от первого всплеска вдохновения до выпуска и долгосрочной поддержки.

Важное примечание: Эта книга фиксирует наше видение технологий, инструментов и подходов в области ИИ по состоянию на весну 2025 года. Сфера искусственного интеллекта развивается стремительно, поэтому к моменту чтения некоторые модели, их возможности, API и тарифы могут измениться. Мы приложили все усилия, чтобы представить точную картину на момент написания, но рекомендуем всегда сверяться с актуальной информацией на официальных сайтах разработчиков.

1.1. Почему весна 2025 – подходящее время для старта с ИИ

Мир искусственного интеллекта стремительно эволюционирует. К весне 2025 года сформировался достаточно зрелый и разнообразный набор доступных инструментов, объединяющий как глобальных лидеров, так и перспективные локальные российские разработки. От GPT-4 Turbo и Google Gemini до отечественных Sber GigaChat и YandexGPT – многие платформы предлагают уникальные возможности, доступные уже сейчас. При этом индустрия не стоит на месте: постоянно анонсируются новые версии моделей (ожидаются GPT-5, Claude Next, Gemini 2, Qwen 2 и другие), что делает момент удачным для входа в сферу – инструменты уже мощные, но продолжают активно развиваться.

Ключевые факторы привлекательности ИИ на данный момент:

- **Доступность:** Наличие бесплатных квот для старта (free tiers), промо-акций и относительно доступных тарифов для экспериментов и MVP.
- **Разнообразие:** Широкий спектр моделей для решения различных задач – генерация текста, кода, дизайна, поиск и анализ данных, работа с аудио и видео.
- **Интеграция:** Наличие программных интерфейсов (API) и веб-интерфейсов, часто достаточно простых в освоении даже для начинающих.

1.2. Наш подход: «человек + ИИ»

Мы исходим из того, что успех современного digital-продукта зависит от **синергии** профессиональных навыков человека и возможностей искусственного интеллекта. ИИ – это мощный инструмент-помощник и катализатор идей, но он не заменяет эксперта. Успех лежит в синергии: способности человека к критическому мышлению, креативности и пониманию контекста дополняются вычислительной мощностью и скоростью ИИ. В этой книге вы увидите:

- Как использовать ИИ для генерации идей и быстрого прототипирования.
- Как создавать контент (текст, изображения, код) с помощью нейросетей, **сохраняя при этом полный контроль** над качеством и финальным результатом.
- Как организовать эффективный **многосервисный рабочий процесс (workflow)**, объединяя сильные стороны разных ИИ-моделей.
- Как учитывать вопросы надежности, безопасности данных и юридические аспекты при взаимодействии с зарубежными и локальными ИИ-сервисами.

1.3. Как читать книгу

Книга условно разделена на три крупных блока, соответствующих основным этапам погружения в тему:

– **Блок 1: Основы и обзор инструментов (Части I—III):** Здесь мы рассмотрим текущий ландшафт ИИ-моделей, их сравнение, тарифы, способы регистрации и доступа, а также основы взаимодействия с ИИ – промпт-инжиниринг. Это фундамент для дальнейшей работы.

– **Блок 2: Практическое применение (Части IV—IX):** Основная, наиболее практическая часть книги. Мы пошагово пройдем все этапы создания digital-продукта: генерацию кода и дизайна, тестирование, выпуск, поддержку и мониторинг, демонстрируя активное использование ИИ на каждом шаге

– **Блок 3: Дополнительные материалы (Части X—XII):** Кейсы из реальной практики для разных ролей, полезные приложения (шаблоны промптов, фрагменты кода, инструкции), глоссарий и взгляд в будущее ИИ.

Для максимальной эффективности рекомендуем:

– По возможности читайте главы **последовательно**, особенно в первых двух блоках, так как материал часто строится на предыдущих главах.

– **Не бойтесь экспериментировать!** Выполняйте практические задания (если они есть в главе) или применяйте описанные подходы к своим реальным или учебным проектам.

– Используйте готовые **шаблоны промптов и фрагменты кода** из приложений (Часть XI) как отправную точку для своих задач.

– **Следите за актуальной информацией:** Сфера ИИ меняется ежедневно. Читайте популярные сайты про ИИ и разработку (например, Хабр, VC), блоги разработчиков ИИ, профильные Telegram-каналы. Множество статей и обсуждений помогут в реализации ваших идей и поддержании знаний в актуальном состоянии.

«Рекомендации по чтению для разных аудиторий:

Эта книга будет полезна как опытным специалистам в области digital-технологий, так и широкому кругу читателей, стремящихся освоить искусственный интеллект и понять его практическую роль в создании современных продуктов. Чтобы извлечь из книги максимум пользы, вы можете адаптировать чтение под свои задачи и интересы:

– **Разработчикам** для глубокого погружения в технические аспекты стоит уделить особое внимание Частям IV (Код и автоматизация разработки, включая интеграцию через API), VII (Тестирование и валидация MVP – *особенно разделы, связанные с юнит- и интеграционными тестами, а также тестированием AI-компонентов*), IX (Метрики и мониторинг мульти-ИИ-экосистемы) и Приложениям с кодом (Часть XI).

– **Дизайнерам и Маркетологам** для освоения креативных и коммуникационных возможностей ИИ будут особенно интересны Части V (Дизайн, иллюстрации и мультимедиа), VII (Выпуск продукта и монетизация – *особенно раздел 8.2 о продвижении*), X (Практические кейсы) и Приложениям с шаблонами промптов (Часть XI).

– **Продакт-менеджерам и Предпринимателям** для стратегического видения и управления продуктом рекомендуем сосредоточиться на Частях I – III (Обзор ИИ-ландшафта, Доступ, Промптинг), VI (Тестирование и валидация MVP), VII (Выпуск продукта и монетизация), VIII (Поддержка и масштабирование), X (Практические кейсы) и XII (Тренды и будущее ИИ).

– **Всем, кто хочет освоить ИИ** и понять его практическое применение в создании продуктов: начните с Частей I – III, чтобы разобраться в основах – что такое ИИ сегодня, какие инструменты существуют и как с ними эффективно взаимодействовать. Затем погрузитесь в Часть X (Практические кейсы), чтобы увидеть ИИ в действии на реальных задачах, и допол-

ните картину Частью V (чтобы узнать о возможностях ИИ в создании различного контента). Не пропустите разделы об этике (1.4) и будущих трендах (Часть XII) – они помогут сформировать комплексное и ответственное представление о технологии.

Тем не менее, для целостного понимания процесса создания продуктов с ИИ и лучшего взаимодействия со специалистами разных профилей, мы рекомендуем ознакомиться со всеми частями книги.

1.4. Этика и Ответственное использование ИИ

Создание продуктов с использованием искусственного интеллекта – это не только технологический, но и **этический вызов**. По мере того как ИИ становится все более мощным и интегрированным в нашу жизнь, возрастает и **ответственность разработчиков** за его воздействие. Игнорирование этических аспектов может привести не только к репутационным потерям, но и к реальному вреду для пользователей и общества.

В рамках этой книги мы будем периодически касаться этических вопросов в контексте конкретных технологий, но важно с самого начала обозначить **ключевые принципы ответственной разработки с ИИ**:

– **Предвзятость (Bias) и Справедливость (Fairness)**: ИИ-модели обучаются на данных, которые могут отражать существующие в обществе предубеждения (гендерные, расовые, социальные и т.д.). Это может привести к генерации предвзятых результатов или дискриминации определенных групп пользователей. **Важно**: Осознавать этот риск, стремиться использовать разнообразные и репрезентативные данные для обучения (где это возможно), тестировать продукты на предмет несправедливых исходов и предвзятости.

– **Прозрачность (Transparency) и Объяснимость (Explainability)**: Многие ИИ-модели работают как «черные ящики». Понимание того, *почему* ИИ принял то или иное решение, может быть сложным (проблема донесения информации). **Важно**: Стремиться к максимальной прозрачности (например, четко информировать пользователей, когда они взаимодействуют с ИИ, а не с человеком) и использовать методы повышения донесения информации (explainability methods), где это критично и технически возможно (например, в медицине или финансах).

– **Безопасность (Safety) и Надежность (Reliability)**: ИИ может генерировать неточный, вредоносный, неуместный или фактически неверный контент («галлюцинации»). **Важно**: Внедрять механизмы фильтрации контента, модерации, контроля качества, особенно в продуктах, где ошибки ИИ могут иметь серьезные последствия (например, в рекомендательных системах, влияющих на здоровье или финансы).

– **Конфиденциальность (Privacy)**: Использование ИИ часто связано с обработкой больших объемов данных, в том числе персональных. **Важно**: Строго соблюдать законодательство о защите данных (ФЗ-152 в РФ, GDPR в ЕС и др.), обеспечивать безопасность данных при их передаче и хранении, получать необходимые согласия пользователей и быть прозрачным в отношении использования их данных.

– **Ответственность (Accountability)**: Кто несет ответственность за действия ИИ? Этот сложный юридический и этический вопрос часто не имеет простого ответа. **Важно**: Разработчики и компании должны быть готовы нести ответственность за продукты, которые они создают, предвидеть потенциальный вред и внедрять механизмы для его смягчения и компенсации.

– **Влияние на общество**: Создавая продукты с ИИ, стоит задумываться об их более широком социальном воздействии: на рынок труда (автоматизация профессий), на доступность и достоверность информации (фейки, дезинформация), на социальное неравенство.

Мы призываем вас подходить к использованию ИИ не только с точки зрения технологических возможностей, но и с позиции **ответственности** перед вашими пользователями и обществом в целом. Этические соображения должны быть неотъемлемой частью процесса проектирования, разработки и эксплуатации продуктов с ИИ.

Приступим к погружению в мир ИИ-разработки!

Часть I. Полный обзор ИИ-ландшафта на весну 2025

(Примечание: Списки инструментов ниже являются репрезентативными, но не исчерпывающими. Ландшафт ИИ постоянно меняется. Обязательно проверяйте актуальную информацию на официальных сайтах разработчиков!)

2.1. Запущенные модели

Ниже представлен обзор ключевых ИИ-платформ и моделей, актуальных на весну 2025 года.

2.1.1 Основные LLM (Большие языковые модели) и мультимодальные платформы:

- **OpenAI** (GPT-4 Turbo, DALL·E 3, Sora, ChatGPT API)
- **Google** (Gemini family, Vertex AI API)
- **Anthropic** (Claude 3 family: Opus, Sonnet, Haiku)
- **Mistral AI** (Mistral Large, Mistral Small, Mistral 7B, Mixtral)
- **Meta** (Llama 3 family, Llama 2)
- **Alibaba Cloud** (Qwen family)
- **Tencent** (Hunyuan models)
- **xAI** (Grok)
- **Deepseek AI** (Deepseek Coder, Deepseek LLM)
- **Perplexity AI**

2.1.2 Генерация изображений (специализированные сервисы):

- **Midjourney**: midjourney (Основной интерфейс через Discord)
- **Stability AI** (Stable Diffusion модели: SDXL, SD 3 и др.)
- **Ideogram**
- **Leonardo AI**
- **Kandinsky (Sber AI)**: Доступ через GigaChat, API Сбера и telegram-канал.
- **Adobe Firefly**

2.1.3 Специализированные ИИ-сервисы по категориям: *(Примеры направлений, список быстро устаревает)*

– **Видео (Генерация, Редактирование, Аватары)**: Runway, Pika, HeyGen, Synthesia, Captions AI, Kling.ai, VEO (Google), Sora (OpenAI).

– **Аудио (Синтез речи, Музыка, Обработка)**: ElevenLabs, Yandex SpeechKit, Sber Salut, Suno AI, Udio, Adobe Podcast Enhance.

– **Дизайн и Презентации**: Gamma, Tome, Designs, Microsoft Designer, Uizard.

– **Разработка и Автоматизация**: GitHub Copilot, Tabnine, Cody, v0.dev, Make, Zapier.

– **Контент и Маркетинг**: Jasper, Copy, Writesonic.

2.1.4 Другие API и платформы:

– **Hugging Face**: Хаб моделей, датасетов, инструментов, API.

– **Replicate**: Платформа для запуска множества AI-моделей через API.

– **DeepAI**: API для различных ИИ-задач.

2.2. Российские решения

- **Yandex Cloud** (YandexGPT API, Yandex SpeechKit, Yandex Vision и др.)
- **Sber AI** (GigaChat, Kandinsky, SaluteSpeech, SmartMarket API)
- **Tinkoff AI**
- Информация: ИИ-решения интегрированы в продукты Тинькофф (скоринг). Публичные API ограничены.

2.3. Анонсы ближайших релизов (Ожидания на конец 2025 – 2026) (С высокой степенью неопределенности)

– **Новые поколения моделей:** GPT-5, Claude Next, Gemini 2, Llama 4 и др. (улучшенная производительность, рассуждение, мультимодальность, контекст).

– **Развитие AI-агентов:** Более автономное выполнение сложных задач.

– **On-Device AI:** Рост числа моделей для локальной работы на устройствах.

– **Улучшение генерации видео:** Прогресс в качестве, управляемости, длительности.

– **Регулирование:** Новые законодательные инициативы в области ИИ.

Примечание: Сроки и характеристики являются предположениями.

2.4. Сравнительная матрица по ключевым задачам

В таблице ниже приведена **очень ориентировочная** сравнительная оценка некоторых популярных моделей на **весну 2025 года**. **Обязательно проводите собственное тестирование!**

Задача	Текст (качество генерации)	Код (точность, сложность)	Дизайн (качество изображений, мультимодальность)*	Анализ данных (структурирование, выводы)	Цена за API (относительная*)
GPT-4 Turbo (OpenAI)	★★★★★	★★★★★	★★★★☆ (с DALL·E)	★★★★☆	\$\$\$ (Высокая)
Gemini Pro (Google)	★★★★☆	★★★★☆	★★★★☆	★★★★☆	\$\$\$ (Высокая)
Claude 3 Opus (Anthropic)	★★★★★	★★★★☆	★★★☆☆ (базовая мультимодальность)	★★★★☆	\$\$\$ (Высокая)
Qwen (Tencent)	★★★★☆	★★★★☆	★★★☆☆	★★★☆☆	\$\$ (Средняя)
Llama 3 (Meta)	★★★★☆	★★★★☆	★☆☆☆☆	★★★☆☆	N/A (Open Source) / \$\$ (API)
Grok (xAI)	★★★☆☆	★★★☆☆	★☆☆☆☆	★★★☆☆	\$\$ (Средняя)
GigaChat Pro (Sber)	★★★☆☆	★★★☆☆	★★★☆☆ (с Kandinsky/RuDALL·E)	★★★☆☆	\$ (Низкая)
YandexGPT (Yandex)	★★★☆☆	★★★☆☆	★☆☆☆☆	★★★☆☆	\$ (Низкая)

Легенда и примечания:

Оценка: ★★★★★ – Отлично, ★★★★☆ – Хорошо, ★★★☆☆ – Удовлетворительно, ★★☆☆☆ – Посредственно, ★☆☆☆☆ – Слабо / Неприменимо.

*Текстовые модели (Llama, Grok, YandexGPT) не предназначены для генерации изображений, поэтому их оценка в столбце «Дизайн» минимальна.

****Цена API:** \$ – Низкая, \$\$ – Средняя, \$\$\$ – Высокая (относительная оценка стоимости использования API за объем токенов/запросов). Для Open Source моделей цена API зависит от хостинга.

Важно! Эта таблица представляет собой субъективную экспертную оценку, актуальную на весну 2025 года. Возможности моделей и их сравнительная производительность могут быстро меняться. Используйте эту таблицу как отправную точку, но всегда проводите собственное тестирование для ваших конкретных задач и проверяйте актуальные тарифы на официальных сайтах.

2.5. Типы аккаунтов и тарифы

Провайдеры ИИ обычно предлагают несколько уровней доступа:

- **Личные бесплатные аккаунты (Free Tier):** Базовый доступ с существенными ограничениями. Для знакомства и простых экспериментов.
- **Премиум-подписки для частных лиц (Premium/Plus/Pro):** Платный доступ (\$10—20/мес.) к продвинутым моделям и функциям в веб-интерфейсах (не API).
- **Платное использование API (Pay-as-you-go):** Основная модель для разработчиков. Оплата по факту за токены, запросы, изображения и т. д. Тарифы сильно различаются.
- **Корпоративные тарифы (Enterprise):** Для бизнеса. Включают высокие лимиты, SLA, fine-tuning (иногда), расширенную безопасность/compliance, поддержку, договорные цены.

Часть II. Начало работы с ИИ

После детального обзора современного ИИ-ландшафта и ключевых технологий в Части I, настало время перейти к практическим аспектам работы с искусственным интеллектом. В этой части книги, мы подробно рассмотрим два основных пути взаимодействия с ИИ-инструментами:

- **Непосредственное использование через готовые интерфейсы:** Мы изучим популярные веб-сайты и мобильные приложения, которые позволяют применять возможности многих из упомянутых ранее платформ без необходимости программирования. Это зачастую самый быстрый способ начать работу и получить первые результаты (раздел 3.1).

- **Интеграция через API (Application Programming Interface):** Мы погрузимся в технические детали подключения ИИ-сервисов к вашим собственным digital-продуктам, что открывает безграничные возможности для кастомизации и создания уникальных функций (разделы 3.3 – 3.5).

Особое внимание в этой части будет уделено актуальным для пользователей из России вопросам: легальным и практическим способам доступа к зарубежным и отечественным ИИ-сервисам (раздел 3.2), нюансам оплаты и фундаментальным принципам безопасного управления API-ключами (раздел 3.5).

3.1. Веб- и мобильные интерфейсы (без использования API)

Этот способ подходит для быстрого тестирования идей, генерации контента вручную, обучения и решения повседневных задач без необходимости программирования.

3.1.1 ChatGPT (OpenAI)

- **Веб-версия:** chat.openai

- **Мобильные приложения:** iOS / Android

- **Функционал:** Текстовый чат, доступ к разным версиям GPT (GPT-4o бесплатно, GPT-4 по подписке), кастомные инструкции (Custom Instructions), GPTs (пользовательские боты в платной версии).

- **Платный доступ:** ChatGPT Plus/Team/Enterprise (~\$20/мес. за Plus на весну 2025) – доступ к последним моделям (GPT-4, DALL-E 3, Advanced Data Analysis и др.).

3.1.2 Google Gemini

- **Веб-версия:** gemini.google

- **Интеграция:** В сервисах Google Workspace (Docs, Sheets и т.д.) с подпиской Google One AI Premium или корпоративной.

- **Примеры использования:** Генерация текста, ответы на вопросы, помощь с кодом, анализ данных, работа с изображениями.

3.1.3 Anthropic Claude

- **Веб-версия:** claude

- **Тарифы:** Бесплатный уровень с ограничениями, платные подписки (Claude Pro ~\$20/мес. на весну 2025) для снятия лимитов и доступа к последним моделям (Opus, Sonnet).

3.1.4 Qwen (Alibaba Cloud)

- **Веб-версия:** Интерфейс Tongyi Qianwen (tongyi.aliyun/qianwen/) или через консоль Alibaba Cloud.

- **Доступность:** Интерфейс может быть на китайском/английском.

3.1.5 Grok (xAI)

- **Интеграция:** Доступен подписчикам X Premium+ в интерфейсе платформы X.

3.1.6 Российские веб-инструменты

- **GigaChat (Sber):** Веб-чат gigachat.sber или через developers.sber/gigachat/login (требуется Sber ID).

- **YandexGPT:** Интегрирован в сервисы Яндекса (Поиск, Браузер, Алиса, Шедеврум).

3.1.7 Встроенный ИИ в популярных приложениях

- **GitHub Copilot:** Помощь с кодом в IDE (VS Code, JetBrains). Требуется подписка.

- **Adobe Creative Cloud:** Функции на базе Adobe Firefly (Generative Fill). Подписка Adobe CC.

- **Canva:** Инструменты Magic Write, Magic Media и др. Free/Pro версии.

- **Google Workspace:** Функции с ИИ при наличии подписки (Duet AI / Google One AI Premium).

- **Microsoft 365 Copilot:** Интеграция ИИ в Office. Отдельная подписка.

3.1.8 Боты и каналы в Telegram и Discord

- **Telegram-боты:** Существует множество неофициальных ботов, предоставляющих доступ к API разных моделей (GPT, Midjourney, Stable Diffusion, YandexGPT, GigaChat и др.).

- **Риски:**

- **Безопасность:** Не передавайте конфиденциальные данные через таких ботов. Неизвестно, кто и как обрабатывает ваши запросы.

- **Легальность:** Часто нарушают условия использования официальных API.

- **Стабильность:** Могут перестать работать в любой момент.
- **Актуальность:** Могут использовать старые версии моделей.
- **Стоимость:** Часто требуют подписки или имеют неочевидные платежи.
- *Примеры (могут быть неактуальны):* @ChatGPT_RU_bot, @genimg_bot, @yandexgpt_bot, @kandinsky21_bot, @gigachat_bot.
- **Discord-серверы:** Многие ИИ-проекты (особенно Midjourney, Stable Diffusion) используют Discord как основную платформу для взаимодействия с ИИ через команды ботам в специальных каналах. Здесь обычно больше контроля и официальной поддержки, чем в случайных Telegram-ботах.
- **Интеграции:** Некоторые платформы (CRM, таск-трекеры) позволяют подключать ИИ-ботов через **webhooks** (веб-хуки – уведомления о событиях) или API к своим чатам для автоматизации задач (например, суммирование переписки, генерация ответа).

3.2. Интеграция через API: квоты и тарифы

API позволяет встраивать ИИ в ваши продукты.

3.2.1 Бесплатные API и квоты (Free Tiers)

Многие провайдеры предлагают **бесплатные кредиты** или **лимиты** для тестирования API.

Сервис	Бесплатная квота	Срок
OpenAI	\$18 кредита (~40 000 токенов)	3 мес.
Google Gemini	\$300 кредита	90 дн.
Anthropic	50 000 токенов	тестовый период
Qwen	10 000 запросов	без ограничения
Grok	5 000 запросов	30 дн.
Deepseek	5 000 документов	по приглашению
SberAI	до 100 000 токенов	пилотные проекты
YandexGPT	до 50 000 токенов	пилотные проекты

*Примечание: Условия бесплатных квот **очень динамичны**. Всегда проверяйте актуальную информацию на сайтах провайдеров.*

3.2.2 Профессиональные тарифы и платный API (Pay-as-you-go)

Основной способ – **оплата по мере использования**.

– **Принцип:** Оплата за **объем потребленных ресурсов** (токены входа/выхода, запросы, изображения и т.д.).

– **Цены:** Сильно различаются у провайдеров и моделей.

– **Примеры ценовых моделей (ОЧЕНЬ ориентировочные, весна 2025):**

– *OpenAI API:* Тарифы за 1 млн токенов (МТОК). GPT-4 Turbo ~\$10—60/МТОК. DALL-E 3 ~\$0.04/изображение.

– *Google Cloud AI API (Vertex AI):* Тарифы за 1000 символов/токенов. Gemini Pro ~\$0.1-\$0.5 / 1М токенов.

– *Anthropic API (Claude):* Тарифы за 1 млн токенов. Opus ~\$15—75/МТОК. Sonnet/Haiku дешевле.

– *Другие API (Qwen, Mistral и т.д.):* Своя модель ценообразования.

– **Важно:** Цены **постоянно меняются**. Обязательно изучайте официальные прайс-листы.

3.3. Примеры использования API при разработке

- Генерация уникальных описаний товаров.
- Создание персонализированных рассылок.
- Разработка чат-бота поддержки.
- Автогенерация unit-тестов / mock-данных.
- Интеграция генерации изображений.
- Анализ и суммаризация текстов. *(Более подробно – в Частях IV – IX).*

3.4. Управление ключами и безопасностью API

Безопасность API-ключей – критически важна!

- **Хранение ключей: НИКОГДА** в коде или git! Используйте:
- **Переменные окружения (environment variables):** Самый распространенный способ для приложений.
- **Системы управления секретами (Secrets Management):** HashiCorp Vault, AWS Secrets Manager, Google Secret Manager, Yandex Lockbox, Doppler, KeePass, Azure Key Vault и др. Рекомендуется для сложных систем и команд.
- **Файлы конфигурации: С ограниченными правами доступа,** не коммитьте их в Git (добавляйте в .gitignore). Менее безопасно, чем первые два способа.
- **Ограничения доступа:**
- **Ограничение по IP (IP Whitelisting):** Если API провайдер поддерживает, разрешайте доступ только с IP-адресов ваших серверов.
- **Разделение ключей:** Используйте **отдельные ключи** для разных окружений (dev, test, prod) и разных микросервисов/приложений. Это позволяет точно отзывать скомпрометированные ключи и контролировать использование.
- **Лимиты использования:** Установите **жесткие лимиты** (rate limits по частоте запросов, бюджеты по расходам) в настройках аккаунта у API-провайдера. Это защита от ошибок, атак и утечек ключей.
- **Минимальные права:** Если провайдер позволяет гранулированную настройку прав для ключей (например, доступ только к определенным моделям или функциям), используйте принцип минимально необходимых привилегий.
- **Мониторинг использования:**
- **Регулярно отслеживайте** расходы и количество запросов через дашборды провайдера (см. 10.1).
- **Настройте оповещения (alerting)** на превышение бюджетов/лимитов и аномальный рост использования (см. 10.1).
- **Логируйте** все API-вызовы на своей стороне для аудита и анализа.
- **Ротация ключей:** Периодически меняйте (ротите) API-ключи, особенно если есть подозрение на компрометацию или если ключ используется давно.

Часть III. Prompt-инжиниринг и мульти-ИИ-воркфлоу

Дисклеймер: ИИ – мощный инструмент, но он **не читает мысли** и **не заменит** ваш опыт. Качество результата напрямую зависит от качества вашего запроса (промпта). Не ждите идеального ответа с первой попытки: часто требуются эксперименты, итерации и обязательная проверка результата человеком. Формулируйте четкие, подробные и недвусмысленные запросы.

В этой главе разберем **ключевые техники составления эффективных промптов** для LLM и покажем, как организовать **рабочий процесс с использованием нескольких ИИ (оркестрация)**. Примеры будут на русском и английском.

4.1. Базовые техники промпт-инжиниринга (Prompt Engineering)

4.1.1 Чёткость и конкретика (Be specific)

Избегайте общих запросов. Чем детальнее вы опишете желаемый результат, тем лучше ИИ вас поймет.

– **Плохо:** «Напиши о продуктивности.»

– **Хорошо (Русский):** «Опиши 5 ключевых техник повышения продуктивности для разработчиков ПО, работающих удалённо. Укажи для каждой техники краткое объяснение и пример применения.»

– **Хорошо (English):** «Describe 5 key productivity techniques for software developers working remotely. For each technique, provide a brief explanation and an example of its application.»

4.1.2 Ролевая постановка задачи (Set the role / Act as...)

Предложите ИИ «примерить» роль эксперта в нужной области. Это помогает модели сфокусироваться на релевантном стиле и знаниях.

– **Русский:** «Ты – опытный UX/UI дизайнер. Предложи структуру и основные элементы интерфейса для мобильного приложения-трекера привычек. Обоснуй свой выбор.»

– **English:** «You are an experienced UX/UI designer. Propose the structure and main interface elements for a mobile habit tracker application. Justify your choices.»

4.1.3 Задание формата ответа (Format the output)

Укажите, в каком виде вы хотите получить ответ: список, таблица, JSON, HTML-код, эссе и т. д.

– **Русский:** «Сравни YandexGPT и GigaChat по 3 основным критериям для задачи написания маркетинговых текстов. Представь ответ в виде таблицы Markdown.»

– **English:** «Compare YandexGPT and GigaChat based on 3 main criteria for the task of writing marketing texts. Present the answer as a Markdown table.»

4.1.4 Предоставление контекста и примеров (Provide context and examples / Few-shot prompting)

Если задача сложная или требует специфического стиля, дайте ИИ примеры или дополнительный контекст в самом запросе.

– **Русский:** «Вот пример хорошего рекламного слогана: [Ваш пример слогана]. Напиши еще 3 слогана в похожем стиле для компании, продающей органический кофе.»

– **English:** «Here is an example of a good advertising slogan: [Your slogan example]. Write 3 more slogans in a similar style for a company selling organic coffee.»

4.1.5 Итеративность и уточнение (Iterate and refine)

Редко удается получить идеальный результат с первого раза. Используйте ответы ИИ как основу и уточняйте запрос в последующих сообщениях.

– **Русский (последующий запрос):** «Спасибо. Теперь сделай предложенные тобой слоганы короче и добавь в них упоминание бесплатной доставки.»

– **English (follow-up prompt):** «Thanks. Now make the slogans you proposed shorter and mention free shipping in them.»

4.2. Оркестрация нескольких моделей (Multi-AI Workflow)

Разные ИИ-модели сильны в разном. Для сложных задач эффективна **комбинация моделей (оркестрация)**. (Помните: конкретные модели устаревают, ориентируйтесь на класс/специализацию).

Пример стратегии:

- **Идея/Структура:** Быстрая/дешевая модель (Haiku, Flash, GigaChat).
- **Проработка:** Мощная модель (GPT-4 Turbo, Opus, Gemini Pro).
- **Специализация:**
 - *Код:* Copilot, Deepseek Coder, GPT-4.
 - *Изображения:* Midjourney, Stable Diffusion, DALL·E 3, Kandinsky.
 - *Анализ/Поиск:* Perplexity, Gemini (с веб-доступом).
 - *Перевод:* DeepL, GPT-4.
 - *RU-контекст:* YandexGPT, GigaChat.

Пример пайплайна для создания статьи (Концептуальный Python-код):

Python

```
# - - Шаг 1: Генерация структуры (быстрая модель) - -
prompt_outline = «Создай структуру статьи „ИИ в малом бизнесе в 2025“»
# response = call_fast_model_api(prompt_outline) # Вызов API
outline = «1. Введение...» # Полученная структура (пример)
# - - Шаг 2: Написание разделов (мощная модель) - -
full_draft =»»
# for section_prompt in parse_outline(outline):
# section_text = call_powerful_model_api(section_prompt) # Вызов API
# full_draft += section_text + "\n»
full_draft = «Текст введения... Текст раздела 1...» # Результат (пример)
# - - Шаг 3: Генерация иллюстрации (модель изображений) - -
prompt_image = «Иллюстрация: график роста эффективности после внедрения ИИ»
# image_url = call_image_model_api(prompt_image) # Вызов API
image_url = "http://example.com/image.jpg" # Полученный URL (пример)
# - - Шаг 4: Редактура человеком - -
final_text = human_edit(full_draft) # Важнейший шаг!
```

Описание: Псевдокод иллюстрирует пайплайн: быстрая модель -> структура (outline), мощная модель -> текст (full_draft), модель изображений -> картинка (image_url), человек -> финальный текст (final_text). Реальные вызовы API (call..._api) опущены.

4.3. Ловушки при мульти-ИИ-подходе (Common Pitfalls)

Потеря контекста: Важно сохранять и передавать контекст между вызовами.

– **Потеря контекста:** При передаче задачи от одной модели к другой важно сохранять и передавать весь необходимый контекст (предыдущие шаги, общую цель).

– **Несогласованность стиля:** Разные модели могут генерировать текст или код в разном стиле. Требуется либо задавать стиль в промпте каждой модели, либо проводить пост-обработку для унификации.

– **Усложнение процесса:** Оркестрация требует настройки и управления. Оцените, действительно ли выигрыш в качестве, оправдывает усложнение по сравнению с использованием одной (пусть и не идеальной во всем) модели.

– **Отслеживание версий и результатов:** При использовании нескольких API важно логировать, какая модель сгенерировала какой фрагмент, с какими параметрами и когда, чтобы можно было анализировать и отлаживать процесс.

4.4. Практические примеры промптов для разработки

– Генерация маркетингового текста (Лендинг):

– **Русский:** «Ты – копирайтер. Напиши текст для первого экрана лендинга мобильного приложения «MindEase» для медитации. Целевая аудитория – занятые профессионалы 25—40 лет. Подчеркни снятие стресса и простоту использования. Добавь призыв к действию «Попробовать бесплатно».»

– **English:** «You are a copywriter. Write the text for the hero section of the landing page for the «MindEase» mobile meditation app. The target audience is busy professionals aged 25—40. Emphasize stress relief and ease of use. Include a call to action «Try for free».»

– Создание тестовых сценариев (Unit-тесты):

– **Русский:** «Напиши 5 unit-тестов на Python с использованием библиотеки pytest для функции `calculate_discount(price, category)`, которая возвращает цену со скидкой. Учти случаи: цена <0 , категория 'premium' (скидка 20%), категория 'standard' (скидка 10%), неизвестная категория (без скидки).»

– **English:** «Write 5 unit tests in Python using the pytest library for the function `calculate_discount(price, category)` which returns the discounted price. Cover cases: price <0 , category 'premium' (20% discount), category 'standard' (10% discount), unknown category (no discount).»

– Описание концепции для дизайнера (Баннер):

– **Русский:** «Опиши концепцию баннера для рекламы вебинара „ИИ для маркетологов“. Цель – регистрация. Визуал: современный, технологичный, возможно, с абстрактными нейронными связями. Ключевые элементы: Заголовок „Маркетинг Будущего с ИИ“, дата вебинара, кнопка „Зарегистрироваться“. Формат: 1200x628px.»

– **English:** «Describe the concept for an ad banner for the webinar «AI for Marketers». Goal: registration. Visual style: modern, techy, perhaps with abstract neural connections. Key elements: Headline «Future-Proof Your Marketing with AI», webinar date, «Register Now» button. Format: 1200x628px.»

Часть IV. Код и автоматизация разработки

В предыдущих частях мы рассмотрели ландшафт ИИ-инструментов (Часть I), включая как глобальные, так и российские решения (YandexGPT, GigaChat и др.), научились получать к ним доступ (Часть II) и освоили базовые принципы составления запросов (Часть III). Теперь мы погрузимся в одну из самых преобразующих областей применения ИИ – разработку программного обеспечения. Эта часть покажет, как искусственный интеллект, становится помощником современного разработчика на всех этапах – от написания кода и интеграции API до автоматизации тестирования и CI/CD.

Наш ключевой принцип **«человек + ИИ»** остается неизменным. ИИ расширяет возможности разработчика, но **не заменяет** его экспертизу и ответственность. Особенно для российского рынка важно понимать, какие задачи можно эффективно решать с помощью отечественных ИИ, а где целесообразнее обратиться к мировым лидерам.

5.1. Генерация кода: от простых скриптов до сложных библиотек

Способность ИИ генерировать код – одна из самых востребованных функций. Модели обучаются на огромных массивах кода и могут помочь в решении различных задач программирования.

5.1.1 Российские ИИ для генерации кода:

На момент весны 2025 года российские модели, такие как **YandexGPT** и **Sber GigaChat**, показывают хорошие результаты в генерации текстового контента и могут быть полезны для следующих задач кодирования:

- Создания **простых скриптов** (особенно если требуется работа с русскоязычными данными или комментариями).
- Генерации **фрагментов кода (сниппетов)** для стандартных задач на популярных языках (Python, JavaScript).
- Помощи в **понимании или рефакторинге** кода с комментариями на русском языке.

Пример промпта для YandexGPT/GigaChat: «Напиши функцию на Python для перевода текста с русского на английский, используя библиотеку 'translate'. Добавь комментарии на русском языке, объясняющие основные шаги.»

Стоит пробовать отечественные модели в первую очередь для несложных задач или задач, тесно связанных с русским языком. Однако для генерации **сложного кода**, работы с **менее распространенными языками** или узкоспециализированными фреймворками их возможностей (на весну 2025) может пока не хватать по сравнению с лидерами.

5.1.2 Глобальные ИИ и специализированные инструменты:

Для более сложных задач кодирования часто эффективнее использовать ведущие мировые модели и инструменты:

- **GitHub Copilot:** Глубоко интегрированный в IDE (VS Code, JetBrains и др.) помощник. Обучен на огромном объеме кода GitHub. Предоставляет контекстно-зависимые подсказки и генерацию кода (от строк до целых функций) в реальном времени. Требуется подписка.

- **OpenAI GPT-4 / GPT-4 Turbo / GPT-5:** Мощные универсальные модели, хорошо справляющиеся с генерацией сложной логики, алгоритмов, кода на разных языках, тестов, документации. Доступны через API и ChatGPT.

- **Anthropic Claude 3 (Opus/Sonnet):** Сильные модели, часто показывающие хорошие результаты в генерации кода, объяснении его работы и рефакторинге. Особенно Opus известен большим контекстным окном. Доступны через API.

- **Другие (Qwen, Llama, Deepseek Coder и т.д.):** Конкурирующие модели со своими сильными сторонами, часто с фокусом на определенных языках или задачах. Многие доступны как Open Source.

5.1.3 Основные сценарии использования ИИ для генерации кода:

Создание шаблонного кода (Boilerplate):

- *Пример промпта:* «Напиши шаблонный код на Python для простого веб-сервера Flask с одной конечной точкой (>/api/data), возвращающей JSON {'status': 'ok'}.»

– Реализация алгоритмов и функций:

- *Пример промпта:* «Реализуй функцию на Java для сортировки списка строк по длине, от самой короткой до самой длинной, используя Stream API.»

– Перевод кода между языками:

- *Пример промпта:* «Переведи эту функцию PHP, использующую curl для GET-запроса, на Python с использованием библиотеки requests, сохранив логику обработки ответа.»

– Генерация фрагментов кода (Snippets):

– *Пример промпта:* «Дай пример кода на C# для асинхронного чтения содержимого веб-страницы по URL с обработкой таймаута.»

– **Помощь в отладке:**

– *Пример промпта:* «Этот JavaScript код [вставить код] должен выводить числа от 1 до 5, но он закичивается. Помоги найти и исправить ошибку. Объясни причину ошибки.»

– **Рефакторинг и оптимизация:**

– *Пример промпта:* «Проанализируй эту функцию Python [вставить длинную функцию с циклами] и предложи варианты рефакторинга для улучшения читаемости и, возможно, производительности (например, использование list comprehensions или map/filter).»

5.1.4 Качество и проверка:

Критически важно: Независимо от того, используете ли вы российскую или зарубежную модель, **сгенерированный ИИ код ТРЕБУЕТ ОБЯЗАТЕЛЬНОЙ ПРОВЕРКИ ЧЕЛОВЕКОМ**. ИИ может генерировать код с:

– Логическими ошибками.

– Уязвимостями безопасности.

– Неэффективной реализацией.

– Несоответствием вашим стандартам кодирования или архитектуре проекта. **Всегда тщательно анализируйте, тестируйте и адаптируйте** полученный код перед использованием в продакшене. **Не доверяйте ИИ слепо.**

5.2. Интеграция через API: примеры на Python и JavaScript с учетом российских платформ

Встраивание возможностей ИИ в ваши продукты через API открывает широкие перспективы кастомизации и автоматизации. Для российского рынка целесообразно в первую очередь рассмотреть API отечественных платформ для определенных задач.

5.2.1 Приоритет российским API (Yandex Cloud, Sber SmartMarket):

– **Работа с русским языком:** Анализ тональности русскоязычных отзывов, классификация обращений в поддержку, генерация контента для российской аудитории, разработка русскоязычных чат-ботов.

– **Данные и контекст:** Потенциально лучшее понимание российских реалий (законы, география, культурный контекст) в некоторых задачах.

– **Доступность и оплата:** Более простой процесс регистрации, получения ключей и оплаты в рублях для российских лиц.

5.2.2 Использование глобальных API (OpenAI, Anthropic, Google и др.):

– **Передовые возможности:** Доступ к самым мощным и универсальным моделям для задач, требующих максимального качества, сложной логики, мультимодальности, работы с большим контекстом.

– **Работа с международной аудиторией:** Если ваш продукт ориентирован не только на РФ.

– **Специализированные модели:** Доступ к моделям, заточенным под конкретные задачи (научные, медицинские, финансовые и т.д.).

5.2.3 Примеры кода для API-интеграции:

Python (YandexGPT API – концептуальный пример):

*Цель примеров ниже – показать **концептуальную структуру** вызова API, а не предоставлять готовый к копированию код. В реальных приложениях **обязательно** используйте безопасное хранение ключей, полноценную обработку ошибок и сверяйтесь с **актуальной документацией** по URL и форматам запросов/ответов.*

Python (YandexGPT API – концепция вызова):

```
Python
import requests
import os # Для доступа к env переменным
def call_yandex_api (prompt_text):
    # КЛЮЧИ И URL ДОЛЖНЫ БЫТЬ АКТУАЛЬНЫМИ!
    API_KEY = os.getenv («YANDEX_API_KEY»)
    FOLDER_ID = os.getenv («YANDEX_FOLDER_ID»)
    API_URL = "https://llm.api.cloud.yandex.net/..." # Заменить!
    headers = {«Authorization»: f"Api-Key {API_KEY}», ...}
    body = {
        «modelUri»: f"gpt:// {FOLDER_ID} /yandexgpt-lite», # Пример
        «messages»: [{«role»: «user», «content»: prompt_text}],
        «completionOptions»: {«maxTokens»: 100} # Пример
    }
    try:
        response = requests.post (API_URL, headers=headers, json=body, timeout=15)
        response.raise_for_status ()
        result = response.json ()
        # КОНЦЕПТУАЛЬНО: Извлечь ответ из result [...]
```

```

return result.get('result', {}) ... ['text'] # Структура может отличаться!
except Exception as e:
print (f"Yandex API Error: {e}") # Заменить на логирование!
return None

```

Описание: Этот пример показывает основные шаги: формирование заголовков headers с ключом API, создание тела запроса body с указанием модели и сообщениями (системным и пользовательским), отправка POST-запроса с помощью requests.post и концептуальный парсинг ответа. В реальном коде необходима полная обработка ошибок и безопасное управление ключами.

JavaScript (Claude API – концепция вызова):

```

JavaScript
async function callClaudeAPI (promptText) {
// КЛЮЧИ, URL, ВЕРСИЯ API И МОДЕЛЬ ДОЛЖНЫ БЫТЬ АКТУАЛЬНЫМИ!

    const API_KEY = process.env.ANTHROPIC_API_KEY; //
    Безопасное получение
    const API_URL = "https://api.anthropic.com/v1/messages";
    const MODEL_NAME = «claude-3-sonnet-...»; // Заменить!
    const headers = {«x-api-key»: API_KEY, «anthropic-version»:
    «...», ...};
    const body = JSON.stringify ({
    model: MODEL_NAME,
    messages: [{«role»: «user», «content»: promptText}]
    // Добавьте другие параметры (max_tokens и т.д.)

});
try {
const response = await fetch (API_URL, {method: «POST», headers, body});
if (!response.ok) throw new Error (`HTTP ${response.status} `);
const data = await response.json ();
// КОНЦЕПТУАЛЬНО: Извлечь ответ из data [...]
return data.content [0].text; // Структура специфична для Claude!
} catch (error) {
console.error («Claude API Error:», error); // Заменить на логирование!
return null;
}
}

```

Описание: Демонстрирует асинхронный вызов Workspace, формирование headers и body, концептуальный парсинг ответа для Claude API. В реальном коде нужна обработка ошибок, таймауты и безопасное управление ключами.

5.2.4 Ключевые моменты при интеграции API:

- **Безопасность ключей:** Храните ключи вне кода (env, secrets managers).
- **Обработка ошибок:** Предусматривайте ошибки сети, API (4xx, 5xx), таймауты, ошибки парсинга. Используйте try-except/try-catch и логирование.
- **Асинхронность:** Используйте async/await для неблокирующих вызовов в нагруженных приложениях.
- **Управление затратами:** Отслеживайте использование, устанавливайте бюджеты и оповещения

– **Rate Limiting:** Реализуйте повторные запросы с задержкой (exponential backoff) при ошибках HTTP 429.

5.3. Авто-тесты и regression suite: помощь ИИ с учетом российских моделей

ИИ может помочь в создании тестов и тестовых данных, ускоряя обеспечение качества ПО.

5.3.1 Использование российских ИИ:

- **Генерация тестов для кода с русским контекстом:** YandexGPT/GigaChat могут лучше понять код с русскими комментариями/данными и предложить релевантные unit-тесты.

- **Генерация тестовых данных для РФ:** Создание списков российских имен, адресов, ИНН, форматов телефонов и т. п.

- *Пример промпта:* «Сгенерируй 10 JSON объектов с тестовыми данными российских пользователей: имя (русское), город (РФ), телефон (+7XXXXXXXXXX), email.»

5.3.2 Использование глобальных ИИ:

Для генерации **сложных unit-тестов**, тестов для **менее распространенных языков** или покрытия **специфических граничных случаев**, глобальные модели (GPT-4, Claude) или специализированные инструменты (Copilot) могут быть эффективнее.

5.3.3 Сценарии применения ИИ в тестировании:

- **Генерация Unit-тестов:**

- *Пример промпта:* «Напиши unit-тест на Java (JUnit 5) для класса „Calculator“, метод 'add (int a, int b) ', покрывая позитивные, негативные числа и ноль.»

- **Генерация тестовых данных (Mock Data):** ИИ может создавать наборы разнообразных данных для тестирования функциональности, производительности или для заполнения баз данных на этапе разработки.

- *Пример промпта:* «Сгенерируй 5 объектов JSON, представляющих тестовые данные для товаров в интернет-магазине. Каждый объект должен включать следующие поля:

- id: уникальный числовой идентификатор (начиная с 1001).

- productName: название товара (строка, например, «Умные часы Series Z»).

- category: категория товара (строка, одна из следующих: «Электроника», «Книги», «Одежда», «Дом и сад», «Спорт»).

- price: цена товара (число с двумя знаками после запятой, в диапазоне от 50.00 до 15000.00).

- quantityInStock: количество на складе (целое число, от 0 до 100).

- isAvailable: доступность для заказа (булево значение, true/false).

- shortDescription: краткое описание товара (строка, 25—40 слов, привлекательное, подчеркивающее 1—2 ключевые особенности). Данные должны быть разнообразными и правдоподобными для каждого поля.»

- **Генерация тестовых сценариев (например, BDD – Behavior-Driven Development):**

- *Пример промпта:* «Напиши сценарий на Gherkin для Cucumber: Фича – Поиск товара. Сценарий – Успешный поиск. Дано: Пользователь на главной странице. Когда: Он вводит „ноутбук“ в строку поиска и нажимает „Найти“. Тогда: Он видит страницу с результатами поиска, содержащую список ноутбуков.»

5.3.4 Ограничения ИИ в тестировании:

- ИИ хорошо помогает с **unit-тестами** и генерацией **данных**.

- Для **интеграционных** и **сквозных (E2E)** тестов его помощь ограничена (требуется понимание взаимодействия систем).

- **Человеческий контроль НЕОБХОДИМ** для верификации корректности и полноты тестов.

5.4. CI/CD-пайплайн с шагами ИИ: приоритет российским ИИ для текстовых задач

Интеграция ИИ в CI/CD (Continuous Integration / Continuous Deployment) может автоматизировать рутинные задачи, например, создание артефактов релиза.

5.4.1 Использование российских ИИ:

Если коммиты, задачи, обсуждения ведутся преимущественно на **русском языке**, использование API **YandexGPT/GigaChat** для автогенерации **changelog** или **черновиков release notes** может дать более качественные и стилистически корректные результаты.

– *Пример промпта для YandexGPT/GigaChat:* «Проанализируй заголовки коммитов [список]. Сформируй краткий список изменений для CHANGELOG.md на русском, сгруппируй по „Новые возможности“ и „Исправления“.»

5.4.2 Использование глобальных ИИ:

Для задач, **менее зависимых от языка** (анализ структуры кода, следование сложным форматам), глобальные модели могут быть предпочтительнее.

5.4.3 Возможные шаги с использованием ИИ в CI/CD:

– **Автогенерация Changelog:** Скрипт в CI получает коммиты, отправляет заголовки в API ИИ, форматирует ответ, добавляет в CHANGELOG.md.

– **Генерация черновика Release Notes:** Аналогично, но с промптом для пользовательского описания изменений.

– **Обновление документации:** Автогенерация docstrings или обновление разделов справки на основе новых функций.

– **Предварительный анализ кода:** Использование ИИ-инструментов для выявления потенциальных проблем до ревью человеком.

– **Комментирование Pull Request'ов:** Автокомментарии с описанием изменений или результатами анализа.

5.4.4 Пример концептуального шага в GitHub Actions:

Цель: Показать идею вызова внешнего скрипта для генерации заметок к релизу.

YAML

```
#.github/workflows/generate_release_notes.yml
name: Generate Release Notes Draft
on:
  release:
    types: [published]
jobs:
  generate:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v4
        with: { fetch-depth: 0 }
      # ... шаги установки зависимостей...
      - name: Get Commit Info (Conceptual)
        id: commits
        run: echo "::set-output name=messages::$ (git log ...)» # Получаем коммиты
      - name: Generate Notes via Script
```

```
env:  
  AI_API_KEY: $ {{secrets.YOUR_AI_API_KEY}}  
  COMMIT_MESSAGES: $ {{steps.commits.outputs.messages}}  
# Главное – вызов вашего скрипта:  
run: python ./scripts/generate_release_notes.py> draft.md  
# ... (опционально) шаг обновления релиза на GitHub...  
Описание: Этот YAML показывает основную идею: шаги  
для получения кода (checkout), установки зависимостей,  
сбора коммитов и вызов вашего Python-скрипта  
(generate_release_notes.py), который и будет обращаться к API ИИ.  
Детали команд и дополнительных шагов (например, обновление  
релиза) опущены для максимальной краткости. Требуется  
создать сам скрипт.
```

5.4.5 Соображения при использовании ИИ в CI/CD:

- **Стоимость:** Вызовы API ИИ могут генерировать расходы.
- **Время выполнения:** Шаги с ИИ могут замедлить пайплайн.
- **Надежность:** Доступность API ИИ влияет на стабильность CI/CD. Нужны fallback-сценарии.
- **Безопасность:** Безопасное хранение ключей API (секреты CI/CD), риски передачи кода/данных во внешние сервисы.

Заключение к Части IV:

ИИ трансформирует разработку ПО. Разумно используя YandexGPT/GigaChat для задач с русским языком/контекстом и глобальные модели для сложных задач, можно повысить производительность. Ключ – **грамотный выбор инструмента и сохранение контроля и критического мышления за человеком.**

Часть V. Дизайн, иллюстрации и мультимедиа

Создание визуально привлекательного и функционального интерфейса и медиаконтента – важнейший этап. ИИ стремительно меняет подходы, предлагая инструменты для ускорения, генерации идей и создания контента. ИИ становится ассистентом дизайнера и контент-мейкера.

Принцип «**человек + ИИ**» ключевой: ИИ генерирует сотни вариантов, но только **человек-дизайнер** (с его вкусом, насмотренностью, пониманием задачи) выбирает лучшее, дорабатывает и обеспечивает соответствие целям. ИИ – кисть и палитра, но художник – человек.

6.1. Создание визуального контента: сравнение Kandinsky, Midjourney, Stable Diffusion, DALL·E

Генерация изображений по тексту (**text-to-image**) – одна из самых впечатляющих технологий ИИ. Рассмотрим основные инструменты (весна 2025).

6.1.1 Российские ИИ для генерации изображений:

- **Sber Kandinsky:** Развитие модели RuDALL-E от Сбера. Kandinsky обычно доступен через веб-интерфейс, платформу GigaChat или API.

- **Возможности:** Генерация изображений по тексту (text-to-image), смешивание изображений, генерация в разных стилях, дорисовывание/расширение изображения (**inpainting/outpainting**).

- **Сильные стороны:** Хорошее понимание запросов на русском языке, включая культурные реалии и специфические термины. Потенциально более простой доступ и интеграция для пользователей из РФ.

- **Ограничения (на весну 2025):** Качество генерации, фотореалистичность и гибкость управления могут несколько уступать топовым мировым аналогам в решении сложных задач.

- **Пример промпта:** «Нарисуй избушку в зимнем лесу в стиле иллюстраций Билибина» или «Логотип для кофейни „Уютный Угол“, минимализм, теплые тона».

6.1.2 Глобальные ИИ-лидеры:

- **Midjourney:** Один из лидеров по художественному качеству и созданию сложных, детализированных изображений с уникальным стилем.

- **Доступ:** Традиционно используется через бота в Discord; возможен веб-интерфейс (проверяйте актуальность).

- **Сильные стороны:** Высокая артистичность, генерация впечатляющих, часто фотореалистичных или стилизованных изображений, активное сообщество.

- **Ограничения:** Может требовать более изощренного промптинга для точного следования запросу по деталям; модель генерации менее «управляема», чем Stable Diffusion.

- **Stable Diffusion:** Популярная модель с открытым исходным кодом. Существует множество версий (SD 1.5, SDXL, SD 3 и т.д.), пользовательских интерфейсов и дообученных моделей (**checkpoints**).

- **Доступ:** Через API (например, Stability AI), платформы (Leonardo AI), веб-интерфейсы (Automatic1111, ComfyUI, Fooocus), которые можно запускать локально (требует мощной видеокарты) или в облаке.

- **Сильные стороны:** Огромная гибкость за счет открытости, множество дообученных моделей под разные стили, поддержка технологий вроде **ControlNet** для точного управления композицией (например, сохранение позы персонажа), возможность локального запуска (конфиденциальность), наличие API.

Конец ознакомительного фрагмента.

Текст предоставлен ООО «Литрес».

Прочитайте эту книгу целиком, [купив полную легальную версию](#) на Литрес.

Безопасно оплатить книгу можно банковской картой Visa, MasterCard, Maestro, со счета мобильного телефона, с платежного терминала, в салоне МТС или Связной, через PayPal, WebMoney, Яндекс.Деньги, QIWI Кошелек, бонусными картами или другим удобным Вам способом.