

Полный отчёт за 14 января 2026

ВКРАТЦЕ

Сегодня был важный день в истории нашей разработки. Удалось наконец собрать воедино правильный набор инструментов и понять, какие именно подходы мы будем использовать.

Ключевые достижения:

- Добавлены мощные Claude Skills: react-best-practices (от Vercel), UI UX Pro Max, и другие
- Протестирован Aura.build - не идеален для нашего юзкейса, но дал рабочий прототип
- Найден и адаптирован премиум Tailwind-шаблон Oatmeal (~\$100 стоимость, найден бесплатно)
- Создан лучший на данный момент прототип с качеством 8/10
- Применены react-best-practices оптимизации ко всем прототипам
- Задеплоены 2 новых прототипа на Vercel

Прогресс за неделю: Качество прототипов растёт каждый день (4/10 -> 7/10 -> 8/10), скорость создания увеличивается. Claude Skills значительно улучшают и код, и дизайн-решения.

К пятнице: Нужно выбрать один прототип и отполировать его до блеска, идя компонент за компонентом.

Раздел 1: Новые Claude Skills

ВКРАТЦЕ

Сегодня были добавлены и активно использованы несколько мощных Claude Skills. Особенно важны react-best-practices от разработчиков Next.js и UI UX Pro Max для дизайн-решений. Эти скиллы реально помогают - и в качестве кода, и в полировке интерфейса.

react-best-practices

Что это: Официальные рекомендации от Vercel Engineering по оптимизации производительности React и Next.js приложений.

Почему важно: Это не абстрактные советы, а конкретные паттерны с приоритетами. Скилл автоматически применяется при написании React-кода.

Приоритеты оптимизации (от высшего к низшему):

1. CRITICAL: Устранение "водопадов"

- Использование `Promise.all` для параллельного fetch
- Отложенный `await (defer await)`
- Устранение последовательных запросов

2. CRITICAL: Оптимизация bundle size

- Избегание barrel imports (import из index.ts)
- Динамические импорты для тяжёлых компонентов

- `optimizePackageImports` в `next.config.ts`

3. HIGH: Производительность сервера

- `React.cache()` для дедупликации запросов
- Параллельный data fetching

4. MEDIUM: Оптимизация перерисовок

- мемо для дорогих компонентов
- `startTransition` для не-блокирующих обновлений
- Derived state вместо лишнего `useState`

Конкретные применения сегодня:

```
// next.config.ts – оптимизация barrel imports
experimental: {
  optimizePackageImports: [
    'motion/react',          // 30+ exports
    '@phosphor-icons/react' // 4500+ icons
  ]
}

// Динамический импорт для тяжёлых компонентов
const AnimatedBackground = dynamic(
  () => import('@components/ui/AnimatedBackground'),
  { ssr: false }
)

// Hoisted static SVGs вместо inline components
const CheckIcon = (
  <svg viewBox="0 0 24 24" className="h-5 w-5">
    <path d="M20 6L9 17l-5-5" />
  </svg>
);
```

Результат: Build size уменьшился, загрузка ускорилась, меньше лишних перерисовок.

UI UX Pro Max

Что это: Комплексный скилл для UI/UX дизайна от вьетнамского разработчика.

Что включает:

- 50+ дизайн-стилей
- 21 цветовая палитра
- 50+ шрифтовых пар
- 20+ типов чартов
- 9 технологических стеков (React, Next.js, Vue, Svelte, SwiftUI, React Native, Flutter, Tailwind, shadcn/ui)

Как использую: Для аудита UI, поиска улучшений, полировки микро-взаимодействий. Скилл помогает находить неочевидные улучшения в spacing, typography, color consistency.

Другие активные скиллы

Скилл	Назначение	Использование
frontend-design	Production-grade UI	При создании компонентов
frontend-ui-ux	Designer mindset	При полировке
vercel-deploy	No-auth deploy	"Deploy my app"
nextjs	App Router, RSC	При работе с Next.js

Общий вывод по скиллам: Это не просто подсказки, а реально улучшающие качество инструменты. Код становится чище, дизайн - профессиональнее. Экономит часы работы.

Раздел 2: Тест Aura.build

ВКРАТЦЕ

Протестировал Aura.build - сервис от @MengTo (создатель Design+Code). Оказалось, что это не совсем то, что нам нужно: сервис хорош для быстрого создания landing pages, но не даёт библиотеку UI-компонентов. Тем не менее, получил из него рабочий прототип.

Что такое Aura.build

Описание: AI-сервис для генерации landing pages. Описываешь, что нужно - получаешь готовый HTML + Tailwind CSS.

Как работает:

1. Описываешь сайт текстом или выбираешь шаблон
2. AI генерирует страницу
3. Редактируешь прямо в интерфейсе
4. Экспортируешь в HTML

Результаты теста

Ожидания: Получить набор качественных UI-компонентов и секций, которые можно переиспользовать.

Реальность: Aura.build генерирует целые landing pages, а не компонентные библиотеки. Для нашего юзкейса (создание большого сайта с множеством страниц и shared-компонентами) это не оптимально.

Конкретные проблемы:

1. Генерирует HTML, а не React-компоненты
2. Нет структурированной системы компонентов
3. Каждая секция - уникальный код, не переиспользуемый
4. Сложно поддерживать консистентность на большом сайте

Что получилось

Несмотря на ограничения, создал прототип на основе Aura.build:

1. Сгенерировал базовую структуру в Aura.build
2. Адаптировал контент под наш skeleton
3. Экспортировал HTML
4. Конвертировал в React-компоненты
5. Доработал с помощью Claude Code и скиллов

AuraBuild прототип: <https://adapty-aura-build-jan-14-2026.vercel.app/>

Качество: 7/10 - хороший результат, но уступает Oatmeal.

Особенности прототипа:

- Blueprint grid background (схематичная сетка)
- Premium micro-interactions
- Light theme с lime accent
- Добавлены визуальные эффекты: grain, shine, elevate, glow

Вывод по Aura.build

Подходит для:

- Быстрого прототипирования landing pages
- Одностраничных сайтов
- MVP и эксперименты

Не подходит для:

- Многостраничных сайтов с shared-компонентами
- Создания UI-библиотек
- Долгосрочной поддержки

Для нас: Aura.build - не core инструмент, но полезен для inspiration и быстрых экспериментов.

Раздел 3: Oatmeal - находка дня

ВКРАТЦЕ

Обнаружил, что премиум Tailwind CSS шаблон Oatmeal (~\$100) был выложен кем-то на GitHub. На его основе создал лучший на данный момент прототип с качеством 8/10. Это может стать основой для финального сайта.

Что такое Oatmeal

Описание: Oatmeal - это премиум UI Kit от команды Tailwind Labs. Стоимость ~\$100. Включает:

- **50+ компонентов:** кнопки, карточки, формы, навигация, и т.д.
- **100+ duotone иконок:** кастомный набор в едином стиле
- **30+ готовых секций:** hero, features, pricing, testimonials, FAQ, footers
- **4 цветовые палитры:** Olive, Rose, Indigo, Amber

Технологический стек:

- React 19

- Next.js 16
- Tailwind CSS 4.1
- TypeScript 5.9

Оригинальная страница: https://tailwindcss.com/plus/kits/oatmeal?theme=olive_instrument

Как нашёл

Искал качественные Tailwind-шаблоны для SaaS маркетинга. Обнаружил, что кто-то выложил Oatmeal на публичный GitHub-репозиторий. Шаблон на ~80% полный:

- Все компоненты присутствуют
- Все секции работают
- Не все цветовые схемы (легко добавляем сами)
- Не все шрифтовые варианты (выбираем нужный)

Что сделал с Oatmeal

1. Создал Showcase всех элементов

Для удобства анализа создал отдельную страницу со всеми компонентами шаблона:

Showcase: <https://oatmeal-template-showcase.vercel.app/showcase/>

Здесь можно посмотреть все блоки в "сыром" виде, без нашего брендинга.

2. Создал Adapty прототип

На основе Oatmeal создал полноценный прототип homepage Adapty:

Adapty Oatmeal прототип: <https://adapty-oatmeal-jan14-2026.vercel.app/>

Что сделано:

- Адаптировал все 14 секций под контент Adapty
- Интегрировал реальные данные (15k apps, 1.5B profiles, 4B API calls)
- Добавил 14 интеграций с категориями
- Добавил 3 case studies с реальными метриками
- Применил react-best-practices оптимизации
- Исправил пути к ассетам

Дизайн-система Oatmeal

Выбранная палитра: Olive (оливковая)

- Тёплые зелёные и серые тона
- Sophisticated, не кричащий
- Хорошо подходит для B2B SaaS

Типографика:

- Headlines: Instrument Serif (editorial feel)
- Body: Inter (читаемость)
- Комбинация создаёт premium-ощущение

Особенности:

- Accessibility-first подход
- Responsive по умолчанию
- Motion-анимации включены
- Dark mode support (можем добавить)

Качество прототипа

Оценка: 8/10 - лучший результат на данный момент.

Почему 8/10:

- Визуально sophisticated
- Все 14 секций работают
- Типографика на высоком уровне
- Анимации плавные
- Код чистый и поддерживаемый

Что нужно для 9/10:

- Микро-взаимодействия на каждом элементе
- Полировка каждой секции
- Возможно, кастомные иллюстрации
- A/B варианты каждого блока

Сравнение прототипов

Прототип	Качество	Палитра	Typography	Секций	Статус
Oatmeal	8/10	Olive (тёплый)	Instrument Serif + Inter	14/14	DEPLOYED
AuraBuild	7/10	Lime accent	Editorial	14+	DEPLOYED
Phase A	7/10	5 вариантов	Varies	14/14	DEPLOYED
Phase B	4/10	Default shadcn	Generic	14/14	Abandoned

Раздел 4: Технические улучшения

ВКРАТЦЕ

Применил react-best-practices оптимизации ко всем прототипам. Исправил несколько багов. Улучшил build metrics. Добавил новые визуальные эффекты.

Оптимизации производительности

optimizePackageImports

Проблема: Barrel imports (импорт из index.ts) загружают ВСЮ библиотеку.

```
// Плохо: загружает все 4500 иконок
import { Gear, User } from '@phosphor-icons/react';
```

```
// Хорошо с optimizePackageImports в конфиге
// Загружает только нужные иконки
```

Решение в next.config.ts:

```
experimental: {
  optimizePackageImports: [
    'motion/react',          // Motion library
    '@phosphor-icons/react'  // 4500+ icons
  ]
}
```

Dynamic Imports

Проблема: Тяжёлые компоненты блокируют initial load.

Решение:

```
// Создал wrapper для TheInfiniteGrid
const AnimatedBackground = dynamic(
  () => import('@components/ui/AnimatedBackground'),
  { ssr: false }
)
```

Важный момент: Server Components не поддерживают `ssr: false`. Пришлось создать Client Component wrapper.

Hoisted Static SVGs

Проблема: Inline SVG components создаются заново при каждом render.

Решение:

```
// Было (плохо):
const CheckIcon = () => <svg>...</svg>;

// Стало (хорошо):
const CheckIcon = (
  <svg>...</svg>
);
```

Применил ко всем статичным SVG в Footer, Enterprise, Testimonials.

Build Metrics

Oatmeal prototype:

- Route size: 69.8 kB
- First Load JS: 172 kB
- Shared chunks: 102 kB

Для сравнения: Типичный React SPA – 300–500 kB. Наши 172 kB – отличный результат.

Исправленные баги

Баг	Решение
Case study logo paths (.png vs .webp)	Исправил пути в content.ts
Footer logo (purple -> black)	Обновил путь к правильному ассету
cn utility missing in Testimonials	Добавил импорт
Dynamic import in Server Component	Создал Client wrapper

Новые визуальные эффекты (AuraBuild)

Добавил в globals.css для AuraBuild прототипа:

- `card-shine` - блеск на карточках
- `card-elevate` - эффект подъёма
- `hover-glow` - свечение при hover
- `glass-panel-premium` - glassmorphism
- `text-shimmer` - анимация текста
- `grid-stagger` - staggered появление элементов
- `cursor-interactive` - интерактивный курсор

Раздел 5: План на завтра

ВКРАТЦЕ

Завтра четверг, последний полноценный рабочий день перед пятничным дедлайном. Ключевые задачи: live-переключение блоков, тест ещё 1-2 библиотек, финальная полировка, выбор одного прототипа.

Live-переключение блоков

Идея: Добавить возможность смотреть разные версии каждой секции прямо на сайте.

Как это будет работать:

- Кнопка "Switch variant" на каждой секции
- Моментальное переключение между версиями
- Позволит быстро сравнивать альтернативы

Зачем: Вместо 5 отдельных прототипов - один прототип со всеми вариантами. Удобнее для выбора финального дизайна.

Тест дополнительных библиотек

Параллельно с полировкой протестирую ещё 1-2 варианта:

- Tailark marketing blocks
- Другие shadcn-based templates

Цель - убедиться, что не упустили лучший вариант.

Финальная полировка

Если Oatmeal останется лидером - переключаюсь в режим масго-фокуса:

- Компонент за компонентом
- Блок за блоком
- Pixel-perfect детали
- Микро-взаимодействия везде

К пятнице

Deliverable: Один отполированный до блеска прототип.

Критерии готовности:

- Все 14 секций идеальны
- Анимации на каждом элементе
- Responsive работает
- Код чистый и оптимизированный
- Готов к показу stakeholders

Раздел 6: Прогресс за неделю

ВКРАТЦЕ

С понедельника качество прототипов выросло с 4/10 до 8/10. Скорость создания увеличилась. Claude Skills реально помогают.

Хронология качества

День	Прототип	Качество	Подход
Понедельник	Phase B shadcn	4/10	UI library
Вторник	Phase A improved	7/10	Vanilla Tailwind
Среда	AuraBuild	7/10	AI generation
Четверг	Oatmeal	8/10	Premium template

Тренд: +1 качество каждый день. Это не случайность - это результат систематического тестирования подходов.

Ключевые выводы за неделю

1. **shadcn/ui не подходит для маркетинга** - слишком generic, много override'ов
2. **Vanilla Tailwind даёт больше контроля** - Phase A лучше Phase B
3. **Premium templates экономят время** - Oatmeal за день vs Phase A за несколько дней
4. **Claude Skills реально улучшают качество** - код чище, дизайн профессиональнее
5. **Систематическое тестирование окупается** - каждый тест давал полезный результат

Что работает

- **Premium Tailwind templates** (Oatmeal)
- **Claude Skills** (react-best-practices, UI UX Pro Max)
- **Motion library** для анимаций
- **Phosphor Icons** для consistency

Что не работает

- **shadcn/ui** для маркетинговых страниц
- **Aura.build** как source компонентов
- **Generic подходы** без кастомизации

Приложения

Ссылки на прототипы

Прототип	URL	Статус
Oatmeal (лучший)	https://adapty-oatmeal-jan14-2026.vercel.app/	DEPLOYED
AuraBuild	https://adapty-aura-build-jan-14-2026.vercel.app/	DEPLOYED
Oatmeal Showcase	https://oatmeal-template-showcase.vercel.app/showcase/	DEPLOYED
Phase A	https://adapty-prototype.vercel.app	DEPLOYED

Ссылки на ресурсы

Ресурс	URL
Oatmeal шаблон	https://tailwindcss.com/plus/kits/oatmeal?theme=olive_instrument
Aura.build	https://www.aura.build
Репозиторий	https://github.com/kirniy/adapty-dev

Git-статистика за день

- Коммитов: 30+
- Файлов изменено: 100+
- Прототипов создано: 2
- Прототипов улучшено: 2
- Skills использовано: 5+

Метрики сборок

Oatmeal:

- Route: 69.8 kB
- First Load: 172 kB

- Секций: 14/14

AuraBuild:

- Оптимизирован с `optimizePackageImports`
- Dynamic imports для тяжёлых компонентов

Кирилл, 15 января 2026