

ЧЕК-ЛИСТ • 2026 V3.1



Технический SEO аудит

Интерактивный чек-лист для полного технического аудита сайта. 120+ пунктов: скорость, индексация, structured data, Core Web Vitals.

Язык: русский • 120+ пунктов • 8 глав

RankBoost

Содержание

Введение	3
Как использовать этот чек-лист	3
Оценка критичности проблем	4

Глава 1. Подготовка к аудиту	5
1.1 Необходимые инструменты	5
1.2 Что нужно до начала аудита	6
1.3 Первичный сбор данных	7
1.4 Формирование отчёта	8
1.5 Чек-лист подготовки	9

Глава 2. Индексация и краулинг	10
2.1 Анализ robots.txt	10
2.2 Sitemap.xml	11
2.3 Canonical теги	12
2.4 Noindex и директивы индексации	13
2.5 Crawl budget	14
2.6 Логи сервера	15
2.7 Дублирование контента	16

Глава 3. Core Web Vitals и производительность	17
3.1 Largest Contentful Paint (LCP)	17
3.2 First Input Delay и Interaction to Next Paint	18
3.3 Cumulative Layout Shift (CLS)	19
3.4 Time to First Byte и FCP	20
3.5 Оптимизация ресурсов	21
3.6 Кэширование и CDN	22
3.7 Серверная оптимизация	23

Глава 4. Мобильная оптимизация	24
4.1 Mobile-first индексация	24
4.2 Адаптивный дизайн	25
4.3 Мобильная юзабилити	26
4.4 AMP и альтернативы	27
4.5 Тестирование на устройствах	28

Глава 5. Структурированные данные	29
5.1 JSON-LD и Schema.org	29
5.2 Основные типы разметки	30
5.3 Rich Snippets	31
5.4 Валидация разметки	32
5.5 Расширенные сценарии	33

Глава 6. Безопасность и HTTPS	34
6.1 SSL-сертификаты	34
6.2 Mixed content	35
6.3 HSTS и security headers	36
6.4 Чек-лист безопасности	37

Глава 7. Международная SEO	38
7.1 Hreflang теги	38
7.2 Геотаргетинг	39
7.3 Языковые метки	40
7.4 Мультирегиональные стратегии	41

Глава 8. Финальный чек-лист	42
8.1 Критические проблемы	42
8.2 Важные проблемы	43
8.3 Рекомендации по оптимизации	44
8.4 Дополнительные пункты	45

ВВЕДЕНИЕ

Как проводить технический SEO аудит

Добро пожаловать в чек-лист технического SEO аудита

**RankBoost Team**

Эксперты по технической оптимизации сайтов

Технический SEO аудит — это комплексная проверка всех технических аспектов сайта, которые влияют на его видимость в поисковых системах. Этот чек-лист содержит более 120 пунктов, organized по 8 основным направлениям технической оптимизации. Каждый пункт включает описание проблемы, способ проверки и рекомендации по исправлению.

Данный чек-лист разработан для специалистов разного уровня подготовки — от начинающих SEO-специалистов до опытных технических экспертов. Материалы актуальны для 2026 года и учитывают последние обновления алгоритмов Google и Яндекс.

Как использовать этот чек-лист

Проходите чек-лист последовательно от главы 1 к главе 8. Отмечайте выполненные пункты с помощью символов ☐. При обнаружении проблем фиксируйте их в отчёте и приоритезируйте по уровню критичности.

Каждая глава посвящена отдельному аспекту технической оптимизации и содержит:

- Подробное описание проверяемых параметров
- Пошаговые инструкции по проверке
- Таблицы с критическими значениями
- Примеры кода и конфигураций
- Чек-листы для быстрой проверки

Оценка критичности проблем

Для эффективной приоритизации исправлений используйте следующую систему оценки критичности. Каждая проблема, обнаруженная в ходе аудита, должна быть классифицирована по одному из четырёх уровней.

Уровень	Обозначение	Описание	Срок исправления
Критический		Блокирует индексацию или ранжирование	Немедленно
Высокий		Значительно влияет на производительность	1-2 недели
Средний		Умеренное влияние на SEO-показатели	1 месяц
Низкий		Рекомендация по улучшению	По плану

Важно

Приоритет проблем может отличаться в зависимости от типа сайта. Для новостных сайтов критичными будут проблемы с индексацией, для интернет-магазинов — с производительностью и структурированными данными.

Метрики успеха

По завершении аудита и исправления всех критических и высокоприоритетных проблем ожидаемый результат включает:

90+ PageSpeed Score	<2.5s LCP	<0.1 CLS	<200ms INP
-------------------------------	------------------------	-----------------------	-------------------------

Регулярный повтор аудита рекомендуется проводить не реже одного раза в квартал, а для крупных сайтов — ежемесячно. После каждого крупного обновления сайта или изменения инфраструктуры необходимо проводить точечную проверку затронутых компонентов.

ГЛАВА 1

Подготовка к аудиту

1.1 Необходимые инструменты

Прежде чем приступить к техническому аудиту, необходимо подготовить полный набор инструментов. Качество аудита напрямую зависит от используемых инструментов и правильности их настройки. Ниже приведён минимальный набор бесплатных и платных инструментов, необходимых для проведения профессионального аудита.

Бесплатные инструменты Google

Google предоставляет набор бесплатных инструментов, которые являются обязательными для любого технического аудита:

Инструмент	Назначение	URL
Google Search Console	Мониторинг индексации, ошибок, производительности	search.google.com/search-console
PageSpeed Insights	Анализ Core Web Vitals	pagespeed.web.dev
Rich Results Test	Проверка структурированных данных	search.google.com/test/rich-results
Mobile-Friendly Test	Проверка мобильной оптимизации	search.google.com/test/mobile-friendly
Schema Markup Validator	Валидация Schema.org разметки	validator.schema.org

Сканирующие инструменты

Для глубокого анализа структуры сайта необходимы специализированные сканирующие инструменты:

1 Screaming Frog SEO Spider

Бесплатная версия позволяет сканировать до 500 URL. Для крупных сайтов необходима лицензия (~£199/год). Инструмент анализирует: title, meta description, canonical, H1-H6, редиректы, ответы сервера, размеры страниц и многое другое.

2 Sitebulb

Альтернатива Screaming Frog с расширенными возможностями визуализации. Поддерживает глубокий анализ внутренних ссылок, кластеризацию контента и автоматическое определение проблем.

3 Ahrefs Site Audit

Облачный инструмент для регулярного мониторинга. Автоматически отслеживает изменения в техническом состоянии сайта и отправляет уведомления о новых проблемах.

Инструменты для анализа производительности

Для анализа скорости загрузки и Core Web Vitals необходимы следующие инструменты:

- **Chrome DevTools** — вкладки Performance и Lighthouse для детального анализа загрузки
- **WebPageTest** — тестирование из разных геолокаций и на разных устройствах
- **Lighthouse CLI** — автоматизация тестирования производительности
- **GTmetrix** — комбинированный анализ скорости с рекомендациями

```
# Установка Lighthouse CLI
npm install -g lighthouse

# Запуск аудита
lighthouse https://example.com --output=html --output-path=./report.html
```

1.2 Что нужно до начала аудита

Перед началом технического аудита необходимо получить доступ к ряду ресурсов и данных. Отсутствие доступа к некоторым из них может существенно ограничить глубину анализа.

Необходимые доступы

- **Google Search Console** — полный доступ к свойству (не только просмотр)
- **Яндекс.Вебмастер** — доступ к панели инструментов

- **Административная панель CMS** — для проверки настроек и плагинов
- **FTP/SFTP или SSH** — для анализа файловой структуры и серверных конфигураций
- **Аналитика** — Google Analytics / Яндекс.Метрика для данных о трафике
- **Доступ к серверу** — для анализа логов и серверных настроек

Совет

Запросите все доступы минимум за 3-5 дней до запланированного аудита. Часто передача доступов занимает больше времени, чем само проведение аудита.

Документация и исходные данные

Для полноценного аудита также полезно иметь следующую документацию:

Документ	Зачем нужен
Карта сайта (в идеале — визуальная)	Понимание структуры и логики сайта
Список основных URL	Приоритетные страницы для проверки
История изменений сайта	Контекст для обнаруженных проблем
Целевая аудитория	Для оценки мобильной оптимизации и геотаргетинга
Бизнес-цели	Для приоритизации рекомендаций

1.3 Первичный сбор данных

Первичный сбор данных — это критически важный этап, определяющий качество всего аудита. На этом этапе собирается базовая информация о состоянии сайта, которая будет использоваться для дальнейшего анализа.

Шаг 1: Сканирование сайта

Запустите полное сканирование сайта с помощью Screaming Frog или аналогичного инструмента. Убедитесь, что сканирование охватывает все основные разделы сайта.

1 Настройка сканера

Установите лимит на количество URL (для бесплатных версий). Настройте User-Agent, задайте глубину сканирования, включите сканирование поддоменов при необходимости. Отключите сканирование robots.txt, если нужен полный обзор.

2 Экспорт данных

Экспортируйте результаты сканирования в Excel/CSV. Основные отчёты: все URL с кодами ответов, title и description, canonical теги, H1-H6 заголовки, изображения без alt-атрибутов, внутренние ссылки.

3 Сравнение с sitemap

Сравните список отсканированных URL с содержимым sitemap.xml. Выявите страницы, отсутствующие в карте сайта, и наоборот — URL из sitemap, недоступные для сканера.

Шаг 2: Сбор данных из Search Console

Экспортируйте следующие отчёты из Google Search Console:

- Отчёт об индексации страниц (все страницы с состоянием)
- Отчёт о покрытии (ошибки и исключения)
- Отчёт о Core Web Vitals (данные за 28 дней)
- Отчёт о ссылках (внутренние и внешние)

- Запросы (топ-1000 запросов с показами и кликами)

Шаг 3: Анализ скорости загрузки

Проведите тестирование скорости загрузки для ключевых страниц сайта:

Страница	PageSpeed (моб.)	PageSpeed (десктоп)	LCP	CLS
Главная	☐	☐	☐	☐
Каталог товаров	☐	☐	☐	☐
Карточка товара	☐	☐	☐	☐
Страница статьи	☐	☐	☐	☐
Контактная страница	☐	☐	☐	☐

1.4 Формирование отчёта

По результатам сбора данных необходимо сформировать структурированный отчёт. Рекомендуемая структура отчёта технического аудита включает следующие разделы:

1 Резюме (Executive Summary)

Краткое описание состояния сайта, ключевые метрики, общее количество найденных проблем по уровням критичности. Рекомендуемый объём — 1-2 страницы.

2 Детальные находки

Описание каждой проблемы с указанием: URL или группы URL, уровня критичности, описания проблемы, влияния на SEO, рекомендации по исправлению.

3 Приоритизированный план действий

Упорядоченный список задач по приоритету с указанием ожидаемого эффекта и трудозатрат. Разделение на: немедленные действия, 短期 задачи, среднесрочные и долгосрочные.

Лучшая практика

При формировании отчёта используйте визуализации: графики, таблицы сравнения до/после, диаграммы распределения проблем по типам. Это значительно повышает восприятие отчёта заказчиком.

1.5 Чек-лист подготовки

Перед началом аудита убедитесь, что все пункты этого чек-листа выполнены:

- ☐ Получен доступ к Google Search Console
- ☐ Получен доступ к Яндекс.Вебмастер
- ☐ Получен доступ к CMS (админка)
- ☐ Получен доступ к серверу (SSH/FTP)
- ☐ Получены данные аналитики (GA/Метрика)
- ☐ Установлен и настроен Screaming Frog (или аналог)
- ☐ Установлен Lighthouse CLI
- ☐ Подготовлен шаблон отчёта
- ☐ Получена карта сайта (структура разделов)
- ☐ Определены приоритетные URL для проверки
- ☐ Согласованы бизнес-цели аудита
- ☐ Определён срок выполнения аудита

Распространённая ошибка

Не приступайте к аудиту без полного набора доступов. Неполные данные приводят к пропуску критических проблем и снижению ценности отчёта.

ГЛАВА 2

Индексация и краулинг

2.1 Анализ robots.txt

Файл robots.txt является первой точкой контакта между поисковым роботом и вашим сайтом. Правильная настройка этого файла критически важна для управления индексацией. Ошибки в robots.txt могут привести к блокировке важных страниц или, наоборот, к индексации служебных страниц.

Структура файла robots.txt

Файл robots.txt размещается в корне домена и содержит инструкции для поисковых роботов. Базовая структура включает директивы User-agent, Allow, Disallow и Sitemap.

```
# robots.txt для типичного сайта
User-agent: *
Disallow: /admin/
Disallow: /cgi-bin/
Disallow: /tmp/
Disallow: /*?sort=
Disallow: /*?filter=

User-agent: Googlebot
Allow: /

Sitemap: https://example.com/sitemap.xml
```

Чек-лист проверки robots.txt

- ☐ Файл доступен по адресу /robots.txt
- ☐ Нет синтаксических ошибок
- ☐ Не заблокированы важные разделы сайта
- ☐ Не заблокированы CSS и JavaScript файлы
- ☐ Указан корректный URL sitemap.xml
- ☐ Для разных User-agent заданы корректные правила
- ☐ Нет конфликтующих директив

- ☐ Файл не превышает 500 КБ

Частая ошибка

Блокировка CSS и JavaScript файлов в robots.txt делает страницы непригодными для рендеринга. Google не сможет правильно отобразить страницу и может понизить её качество.

Проверка блокировки ресурсов

Используйте инструмент «Просмотреть как Googlebot» в Search Console для проверки рендеринга страниц. Убедитесь, что поисковый робот имеет доступ к:

- Всем CSS-файлам (включая файлы стилей фреймворков)
- Всем JavaScript-файлам
- Изображениям, используемым в дизайне
- Шрифтам (если они загружаются из внешних источников)

2.2 Sitemap.xml

Карта сайта (sitemap.xml) помогает поисковым роботам обнаруживать и индексировать страницы. Правильно составленная карта сайта ускоряет индексацию и помогает поисковой системе понять структуру сайта.

Требования к sitemap.xml

Карта сайта должна соответствовать стандарту протокола Sitemaps.org и содержать актуальные URL всех индексируемых страниц:

```
<?xml version="1.0" encoding="UTF-8"?>
<urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9">
  <url>
    <loc>https://example.com/</loc>
    <lastmod>2026-01-15</lastmod>
    <changefreq>daily</changefreq>
    <priority>1.0</priority>
  </url>
  <url>
    <loc>https://example.com/products/</loc>
    <lastmod>2026-01-14</lastmod>
```

```
<changefreq>weekly</changefreq>
<priority>0.9</priority>
</url>
</urlset>
```

Чек-лист sitemap.xml

- ☐ Файл доступен по стандартному пути /sitemap.xml
- ☐ Указан в robots.txt
- ☐ Содержит только канонические URL (без редиректов)
- ☐ Все URL возвращают код 200
- ☐ Размер файла не превышает 50 МБ (50 000 URL)
- ☐ Для крупных网站地图 используется индекс sitemap
- ☐ lastmod обновляется при изменениях на странице
- ☐ Нет URL с noindex или 404/410 кодами
- ☐ Все URL доступны для сканирования (не заблокированы в robots.txt)

Для крупных сайтов

Если сайт содержит более 50 000 URL, используйте sitemap index, который ссылается на несколько файлов sitemap. Каждый подчинённый sitemap не должен содержать более 50 000 URL.

2.3 Canonical теги

Канонический тег (canonical) указывает поисковой системе предпочтительный URL для страницы с дублирующимся или похожим контентом. Правильное использование canonical критически важно для избежания проблем с дублированием контента.

Правила использования canonical

- Canonical должен указывать на доступную (не заблокированную, не 404) страницу

- Страницы с canonical должны быть доступны для индексации
- Canonical URL должен быть абсолютным (с указанием протокола и домена)
- Для страниц пагинации используйте canonical на первую страницу или self-referencing
- При наличии HTTP и HTTPS — canonical должен указывать на HTTPS версию

Таблица канонических тегов

Ситуация	Правильный canonical
Дублирующиеся параметры URL (?sort, ?page)	Canonical на URL без параметров
HTTP и HTTPS версии	Canonical на HTTPS
WWW и non-WWW	Canonical на предпочтительный вариант
Страницы пагинации	Self-referencing или canonical на первую страницу
Мобильная версия (m.example.com)	Canonical на десктопную версию + alternate

Опасная ошибка

Циклические canonical (страница А каноникалит на В, а В на А) могут привести к тому, что поисковая система проигнорирует все canonical на сайте. Проверяйте цепочки canonical при сканировании.

2.4 Noindex и директивы индексации

Директивы noindex позволяют запретить индексацию конкретных страниц, не блокируя их для сканирования. Это полезно для служебных страниц, внутренних результатов поиска, страниц с временным контентом.

Способы установки noindex

- **Meta robots** — `<meta name="robots" content="noindex">` в head документа
- **X-Robots-Tag** — HTTP-заголовок для не-HTML ресурсов (PDF, изображения)

- **Роботы-директивы в robots.txt** — Google поддерживает noindex в robots.txt

Чек-лист проверки noindex

- ☐ Страницы результатов внутреннего поиска — noindex
- ☐ Страницы корзины — noindex
- ☐ Страницы авторизации — noindex
- ☐ Дублирующиеся страницы (если нет canonical) — noindex
- ☐ Временные промо-страницы после окончания акции — noindex
- ☐ Страницы с thin content (мало полезного контента) — noindex

Noindex vs Disallow

Noindex и Disallow работают по-разному. Disallow блокирует сканирование, но страница может быть индексирована через ссылки. Noindex разрешает сканирование, но запрещает индексацию. Для большинства случаев предпочтительнее noindex.

2.5 Crawl budget

Crawl budget — это количество страниц, которые поисковый робот сканирует за определённый период. Для крупных сайтов (более 100 000 URL) оптимизация crawl budget становится критически важной.

Факторы, влияющие на crawl budget

Фактор	Влияние	Как оптимизировать
Скорость ответа сервера	Высокое	Уменьшить TTFB до <200мс
Наличие 4xx ошибок	Среднее	Убрать из sitemap, настроить 410
Наличие редиректов	Среднее	Убрать цепочки редиректов
Качество контента	Высокое	Улучшить контент или noindex
Внутренние ссылки	Высокое	Оптимизировать перелинковку

2.6 Логи сервера

Анализ логов сервера позволяет увидеть, как поисковые роботы ^{实际ually} ^{爬取}акут ваш сайт. Это самый точный способ понять поведение роботов и выявить проблемы с краулингом.

Что искать в логах

- **Частота сканирования** — сколько раз Googlebot посещает разные страницы
- **Коды ответов** — какие страницы возвращают ошибки
- **Глубина сканирования** — насколько глубоко робот заходит в структуру
- **Время ответа** — как долго сервер обрабатывает запросы роботов
- **Потраченный crawl budget** — сколько ресурсов тратится на служебные страницы

```
# Анализ логов с помощью GoAccess
goaccess access.log --log-format=COMBINED -o report.html

# Фильтрация по Googlebot
grep "Googlebot" access.log | awk '{print $7}' | sort | uniq -c | sort -rn
```

2.7 Дублирование контента

Дублирование контента — одна из самых распространённых технических проблем. Поисковые системы предпочитают показывать одну версию дублирующихся страниц, и если не указать явный канонический URL, выбор будет сделан автоматически, что не всегда совпадает с вашими ожиданиями.

Типы дублирования

- **Абсолютные дубли** — одинаковый контент на разных URL
- **Частичные дубли** — значительное перекрытие контента
- **Структурные дубли** — одинаковый контент, организованный по-разному

1 Выявление дублей

Используйте Screaming Frog для поиска страниц с одинаковым title, description, H1. Сравните содержимое страниц с помощью индекса совпадений (similarity index).

2 Анализ причин

Определите причину дублирования: параметры URL, пагинация, HTTP/HTTPS, www/non-www, сессионные идентификаторы, UTM-метки.

3 Выбор решения

Для каждого типа дубля выберите решение: canonical, 301-редирект, noindex, изменение логики URL, robots.txt.

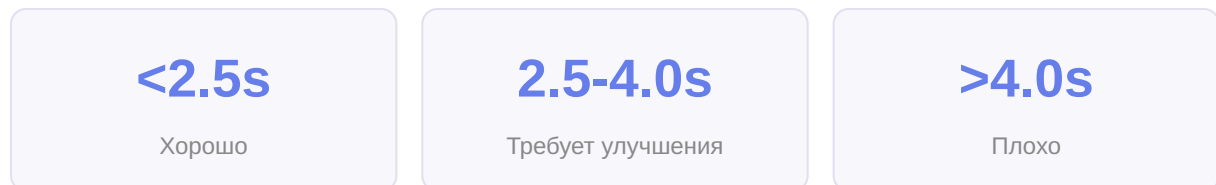
- ☐ Проверены дубли по HTTP и HTTPS
- ☐ Проверены дубли по www и non-www
- ☐ Проверены дубли с параметрами (?sort, ?page, ?filter)
- ☐ Проверены дубли при пагинации
- ☐ Проверены дубли с UTM-метками
- ☐ Проверены дубли с параметрами сессий

ГЛАВА 3

Core Web Vitals и производительность

3.1 Largest Contentful Paint (LCP)

Largest Contentful Paint (LCP) измеряет время загрузки крупнейшего элемента контента в области видимости (viewport). Это один из трёх основных показателей Core Web Vitals, который напрямую влияет на ранжирование в Google.



Типичные элементы LCP

- Изображение или видео в hero-секции
- Крупный баннер или слайдер
- Блок текста, формирующий основной контент
- Элементы с фоновым изображением

Чек-лист оптимизации LCP

- ☐ Оптимизированы изображения (WebP/AVIF формат)
- ☐ Изображения сжаты без потери качества
- ☐ Реализована lazy loading для изображений ниже fold
- ☐ Удалены неиспользуемые CSS и JS файлы
- ☐ Критические CSS инлайнятся в head
- ☐ JavaScript асинхронно загружается
- ☐ Используется preconnect для ключевых доменов
- ☐ Сервер быстро отвечает на запросы (TTFB <200мс)

Ключевой фактор

LCP сильно зависит от времени ответа сервера (TTFB). Если сервер отвечает дольше 200мс, сначала оптимизируйте серверную часть, а затем переходите к клиентской оптимизации.

3.2 First Input Delay и Interaction to Next Paint

First Input Delay (FID) измеряет время между первым взаимодействием пользователя и началом обработки этого взаимодействия. В 2024 году Google заменил FID на более **全面** показатель Interaction to Next Paint (INP), который измеряет время отклика на все взаимодействия.

<100ms

INP — хорошо

100-200ms

INP — улучшить

>200ms

INP — плохо

Причины плохого INP

- Тяжёлый JavaScript, блокирующий основной поток
- Большие обработчики событий
- Длительные операции в цикле **事件循环**
- Частые вызовы reflow/repaint
- Отсутствие yield (деление длинных задач)

Чек-лист оптимизации INP

- ☐ Разбиты длинные задачи (>50мс) на более мелкие
- ☐ Используется requestIdleCallback для не критических задач
- ☐ Debounce обработчики событий (scroll, resize, input)
- ☐ Удалён неиспользуемый JavaScript
- ☐ Code splitting для загрузки только необходимого кода
- ☐ Web Workers для тяжёлых вычислений
- ☐ Ленивая загрузка не критических компонентов

```
# Пример деления длинной задачи
function processLargeArray(items) {
  const chunkSize = 100;
  let index = 0;

  function processChunk(deadline) {
    while (index < items.length && deadline.timeRemaining() > 0) {
      processItem(items[index]);
      index++;
    }
    if (index < items.length) {
      requestIdleCallback(processChunk);
    }
  }
  requestIdleCallback(processChunk);
}
```

3.3 Cumulative Layout Shift (CLS)

Cumulative Layout Shift (CLS) измеряет сумму всех незапланированных сдвигов макета, которые происходят на протяжении всего времени жизни страницы. Высокий CLS указывает на нестабильный визуальный опыт.

<0.1

Хорошо

0.1-0.25

Требуется улучшения

>0.25

Плохо

Основные причины CLS

Причина	Решение
Изображения без атрибутов width/height	Указать явные размеры в HTML или CSS
Рекламные блоки без зарезервированного места	Задать минимальную высоту контейнера
Динамически загружаемый контент	Резервировать место под контент
Шрифты, вызывающие FOUT/FOIT	font-display: swap + preload шрифтов

Причина	Решение
Динамические баннеры	Статические размеры или aspect-ratio

Чек-лист предотвращения CLS

- ☐ Все изображения имеют атрибуты width и height
- ☐ Контейнеры для динамического контента имеют min-height
- ☐ Шрифты загружаются с font-display: swap
- ☐ Внешние виджеты (реклама, чат) имеют стабильные размеры
- ☐ Контент не сдвигается при загрузке шрифтов
- ☐ Используется aspect-ratio для респонсивных элементов
- ☐ Элементы с position: sticky не вызывают сдвигов

Важно

CLS считается на протяжении всего времени жизни страницы, а не только при загрузке. Даже через 30 секунд после загрузки незапланированный сдвиг будет учтён в метрике.

3.4 Time to First Byte и FCP

Time to First Byte (TTFB) измеряет время между запросом пользователя и получением первого байта ответа от сервера. First Contentful Paint (FCP) измеряет время до отрисовки первого элемента контента.

Нормативные значения

Метрика	Хорошо	Улучшить	Плохо
TTFB	<200мс	200-800мс	>800мс
FCP	<1.8s	1.8-3.0s	>3.0s

Оптимизация TTFB

- Использование CDN для доставки контента ближе к пользователю
- Оптимизация серверной логики и базы данных
- Кэширование на уровне сервера (OPcache, Redis, Memcached)
- Использование HTTP/2 или HTTP/3
- Оптимизация TLS (TLS 1.3, OCSP stapling)

Оптимизация FCP

- ☐ Критические CSS инлайнятся в head
- ☐ Удалены блокирующие ресурсы из head
- ☐ JavaScript загружается асинхронно
- ☐ Шрифты загружаются с font-display: optional
- ☐ Сервер отвечает быстро (TTFB <200мс)
- ☐ Используется preconnect к критическим доменам

3.5 Оптимизация ресурсов

Оптимизация ресурсов включает сжатие изображений, кодирование скриптов и стилей, а также правильную настройку доставки ресурсов.

Оптимизация изображений

1 Выбор формата

Используйте WebP как основной формат с fallback на JPEG/PNG. Для поддержки прозрачности — WebP с alpha. Для максимального сжатия — AVIF (с поддержкой ~70% браузеров).

2 Сжатие

Оптимизируйте изображения: JPEG — quality 80-85%, Progressive mode. PNG — quantize colors, remove metadata. WebP — quality 75-80%. Используйте инструменты: imagemin, Sharp, Squoosh.

3 Размеры

Предоставляйте изображения appropriate для размера viewport. Используйте srcset и sizes для респонсивных изображений. Не загружайте изображения 3000px для контейнера 300px.

Оптимизация JavaScript и CSS

- Минификация и сжатие (gzip/brotli)
- Tree shaking для удаления неиспользуемого кода
- Code splitting для загрузки только необходимого
- Deferred loading для некритических скриптов
- Critical CSS extraction

```
# Настройка Brotli сжатия в Nginx
brotli on;
brotli_types text/plain text/css application/json
             application/javascript text/xml application/xml
             image/svg+xml;
```

3.6 Кэширование и CDN

Правильная настройка кэширования значительно ускоряет загрузку страниц для повторных посещений. CDN (Content Delivery Network) обеспечивает быструю доставку ресурсов пользователям из разных географических регионов.

Стратегии кэширования

Тип ресурса	Рекомендуемое время кэширования	Заголовок
Статические ресурсы с хэшем	1 год	Cache-Control: max-age=31536000, immutable
Изображения	1 месяц	Cache-Control: max-age=2592000
HTML страницы	1 час	Cache-Control: max-age=3600
API ответы	0 (без кэширования)	Cache-Control: no-cache

Чек-лист CDN

- ☐ CDN настроен для статических ресурсов
- ☐ Настроены правильные заголовки кэширования
- ☐ Реализовано автоматическое обновление кэша при деплое
- ☐ CDN не блокирует важные API-вызовы
- ☐ Настроены правила очистки кэша (purge)
- ☐ Мониторинг命中率 CDN

3.7 Серверная оптимизация

Серверная оптимизация включает настройку HTTP/2, сжатие, оптимизацию TLS и правильную конфигурацию веб-сервера.

HTTP/2 и HTTP/3

HTTP/2 обеспечивает мультиплексирование запросов, серверный push и сжатие заголовков. HTTP/3 (QUIC) дополнительно устраняет проблемы head-of-line blocking и ускоряет установку соединения.

```
# Настройка HTTP/2 в Nginx
listen 443 ssl http2;
```

```
# Оптимизация TLS
ssl_session_timeout 1d;
ssl_session_cache shared:SSL:10m;
ssl_session_tickets off;
ssl_protocols TLSv1.2 TLSv1.3;
```

Чек-лист серверной оптимизации

- ☐ HTTP/2 включён и работает
- ☐ Сжатие gzip/brotli включено
- ☐ TLS 1.3 поддерживается
- ☐ Keep-Alive включён
- ☐ Оптимизированы настройки буфера
- ☐ Настроена правильная обработка статических файлов
- ☐ Отключены ненужные модули сервера

ГЛАВА 4

Мобильная оптимизация

4.1 Mobile-first индексация

С 2021 года Google использует mobile-first индексацию — поисковая система индексирует и ранжирует мобильную версию сайта. Это означает, что мобильная версия должна содержать весь важный контент и метаданные.

Требования mobile-first индексации

- Мобильная версия должна содержать тот же контент, что и десктопная
- Мета-теги (title, description, canonical, robots) должны присутствовать в мобильной версии
- Структурированные данные должны быть в мобильной версии
- Внутренние ссылки должны работать в мобильной версии
- Hreflang теги должны присутствовать в мобильной версии

Критическая ошибка

Если мобильная версия не содержит контента, который есть на десктопной, этот контент не будет проиндексирован. Проверьте через «Просмотреть как Googlebot» в Search Console.

Чек-лист mobile-first

- ☐ Мобильная версия содержит весь контент десктопной
- ☐ Все мета-теги присутствуют в мобильной версии
- ☐ Структурированные данные размещены в мобильной версии
- ☐ Внутренние ссылки доступны и работают
- ☐ Внешние ссылки корректно работают
- ☐ Hreflang теги настроены для мобильной версии

4.2 Адаптивный дизайн

Адаптивный дизайн — это подход, при котором сайт автоматически подстраивается под размер экрана пользователя. Это рекомендуемый Google способ создания мобильных версий сайтов.

Принципы адаптивного дизайна

1 Viewport meta tag

Убедитесь, что на всех страницах присутствует тег: `<meta name="viewport" content="width=device-width, initial-scale=1">`. Без этого тега мобильные браузеры не будут правильно масштабировать страницу.

2 Гибкая сетка

Используйте relative единицы измерения (% , vw , rem) вместо absolute (px). Flexbox и CSS Grid позволяют создавать гибкие макеты, которые адаптируются к любому размеру экрана.

3 Адаптивные изображения

Используйте srcset для предоставления разных версий изображений для разных размеров экрана. CSS свойство object-fit позволяет контролировать отображение изображений в контейнерах.

Медиа-запросы

```
/* Базовые медиа-запросы */
@media (max-width: 768px) {
  /* Мобильные стили */
  .container { padding: 16px; }
  .sidebar { display: none; }
}

@media (min-width: 769px) and (max-width: 1024px) {
  /* Планшетные стили */
  .grid { grid-template-columns: repeat(2, 1fr); }
}
```

```
@media (min-width: 1025px) {
  /* Десктопные стили */
  .grid { grid-template-columns: repeat(3, 1fr); }
}
```

4.3 Мобильная юзабилити

Помимо технической оптимизации важно обеспечить удобство использования на мобильных устройствах. Google оценивает не только скорость загрузки, но и общий пользовательский опыт.

Проверка мобильной юзабилити

Параметр	Требование	Как проверить
Размер шрифта	Минимум 16px для основного текста	Chrome DevTools → Elements
Размер кнопок/ссылок	Минимум 48x48px область касания	Замер в DevTools
Расстояние между элементами	Минимум 8px между интерактивными элементами	Визуальная проверка
Горизонтальная прокрутка	Отсутствует	Lighthouse, ручная проверка
Размер viewport	Не менее 320px	Media query 检查

Чек-лист мобильной юзабилити

- ☐ Размер шрифта основного текста $\geq 16px$
- ☐ Интерактивные элементы $\geq 48x48px$
- ☐ Нет горизонтальной прокрутки
- ☐ Контент 适配 viewport (нет zoom)
- ☐ Формы удобно заполнять на мобильном
- ☐ Навигация доступна одним пальцем

- ☐ Всплывающие окна не перекрывают контент
- ☐ Скролл работает плавно

4.4 AMP и альтернативы

Accelerated Mobile Pages (AMP) — это фреймворк для создания быстрозагружаемых мобильных страниц. Хотя Google больше не отдаёт явного предпочтения AMP в поиске, технология всё ещё используется для обеспечения мгновенной загрузки.

Когда AMP оправдан

- Новостные сайты с высоким трафиком из Google Discover
- Сайты с контентом, который критичен для скорости (новости, метео)
- Сайты в регионах с медленным мобильным интернетом

Альтернативы AMP

Технология	Преимущества	Ограничения
PWA	Офлайн-режим, push-уведомления	Сложная разработка
Instant Loading	Google определяет автоматически	Ограниченная поддержка
Standard HTML	Полный контроль	Требуется ручная оптимизация

4.5 Тестирование на устройствах

Лабораторные тесты не всегда отражают реальный пользовательский опыт. Важно тестировать сайт на реальных устройствах и в реальных сетевых условиях.

Методы тестирования

1 Эмуляция в DevTools

Chrome DevTools позволяет эмулировать разные устройства, сети и геолокации. Это быстрый способ базовой проверки, но не заменяет тестирование на реальных устройствах.

2 Лабораторное тестирование

Используйте WebPageTest для тестирования с разных локаций и на разных устройствах. Установите ограничение сети (3G, 4G) для проверки в условиях медленного интернета.

3 Полевое тестирование

Собирайте реальные данные CrUX (Chrome User Experience Report) через PageSpeed Insights. Анализируйте данные за 28 дней для стабильных метрик.

Чек-лист тестирования

- ☐ Проверено на iOS (iPhone, iPad)
- ☐ Проверено на Android (разные размеры экрана)
- ☐ Проверено на маленьком экране (320px)
- ☐ Проверено на большом экране (414px+)
- ☐ Проверено в landscape.orientation
- ☐ Проверено на медленном соединении (3G)
- ☐ Проверено с выключенным JavaScript
- ☐ Проверено со включенным блокировщиком рекламы

ГЛАВА 5

Структурированные данные

5.1 JSON-LD и Schema.org

Структурированные данные — это стандартизированный формат представления информации о странице и её содержанием. Google, Яндекс и другие поисковые системы используют структурированные данные для понимания контента и генерации расширенных результатов поиска (rich snippets).

Форматы структурированных данных

Формат	Рекомендация	Особенности
JSON-LD	Рекомендуется	Отдельный блок в head, легко редактируется
Microdata	Допустимо	Встраивается в HTML, сложнее поддерживать
RDFa	Допустимо	Мало используется, сложный синтаксис

Пример JSON-LD

```
<script type="application/ld+json">
{
  "@context": "https://schema.org",
  "@type": "Product",
  "name": "iPhone 16 Pro",
  "image": "https://example.com/iphone16pro.jpg",
  "description": "Смартфон Apple iPhone 16 Pro 256GB",
  "brand": {
    "@type": "Brand",
    "name": "Apple"
  },
  "offers": {
    "@type": "Offer",
    "price": "129990",
    "priceCurrency": "RUB",
    "availability": "https://schema.org/InStock"
  }
}
</script>
```

5.2 Основные типы разметки

Schema.org определяет сотни типов разметки. Ниже приведены наиболее распространённые типы, которые используются на большинстве сайтов.

Типы для разных категорий сайтов

Тип сайта	Рекомендуемые типы Schema.org
Интернет-магазин	Product, Offer, AggregateRating, Review, BreadcrumbList
Новостной сайт	NewsArticle, BreadcrumbList, Organization, WebSite
Блог	BlogPosting, BreadcrumbList, Author, WebSite
Корпоративный сайт	Organization, LocalBusiness, ContactPage, WebSite
Сервисный сайт	Service, FAQPage, HowTo, AggregateRating

Обязательные поля для Product

- ☐ name — название продукта
- ☐ image — изображение продукта (минимум 1)
- ☐ description — описание продукта
- ☐ offers — информация о предложении (цена, наличие)
- ☐ brand — бренд продукта
- ☐ sku — артикул
- ☐ aggregateRating — рейтинг (если есть отзывы)

5.3 Rich Snippets

Rich Snippets — это расширенные результаты поиска, которые отображаются с дополнительной информацией: рейтингами, ценами, доступностью, FAQ и т.д. Они значительно повышают кликабельность (CTR) результатов поиска.

Типы Rich Snippets

Тип	Отображаемая информация	Требуемая разметка
Рейтинги	Звёзды, количество отзывов	AggregateRating
Цены	Цена, валюта, доступность	Offer
FAQ	Вопросы и ответы	FAQPage
How-to	Шаги инструкции	HowTo
Хлебные крошки	Навигационный путь	BreadcrumbList
Видео	Миниатюра, длительность	VideoObject

Потенциал

Сайты с правильной разметкой Product и AggregateRating получают в среднем на 30% больше кликов по сравнению с результатами без rich snippets.

5.4 Валидация разметки

Неправильная разметка может не только не дать ожидаемого эффекта, но и навредить ранжированию. Обязательно проверяйте разметку перед публикацией.

Инструменты валидации

- **Google Rich Results Test** — проверка на возможность появления rich snippets
- **Schema Markup Validator** — проверка синтаксиса Schema.org
- **Google Search Console** — мониторинг ошибок разметки

Чек-лист валидации

- ☐ Валидация пройдена без ошибок в Rich Results Test
- ☐ Валидация пройдена в Schema Markup Validator
- ☐ Нет предупреждений о пропущенных обязательных полях
- ☐ Все URL в разметке доступны (не 404)

- ☐ Изображения в разметке доступны и корректно отображаются
- ☐ Цены и наличие актуальны
- ☐ Разметка соответствует контенту страницы

5.5 Расширенные сценарии

Для сложных сайтов может потребоваться более *avanzada* разметка: динамические данные, разметка для SPA, многоязычная разметка.

Динамическая разметка

Для страниц, данные которых генерируются динамически (каталоги товаров, SearchResult), разметка должна обновляться вместе с контентом:

```
# Серверная генерация JSON-LD
def generate_product_schema(product):
    return {
        "@context": "https://schema.org",
        "@type": "Product",
        "name": product.name,
        "image": product.image_url,
        "offers": {
            "@type": "Offer",
            "price": str(product.price),
            "priceCurrency": "RUB",
            "availability": "https://schema.org/InStock"
            if product.in_stock else
            "https://schema.org/OutOfStock"
        }
    }
```

Разметка для SPA (Single Page Application)

Для SPA-приложений структурированные данные должны генерироваться на стороне сервера (SSR) или через **预渲染**. Google способен рендерить JavaScript, но с задержкой.

Рекомендация

Для критичных страниц (главная, каталог, карточки товаров) используйте SSR для разметки. Для остальных страниц допустима клиентская генерация, но с учётом задержки индексации.

ГЛАВА 6

Безопасность и HTTPS

6.1 SSL-сертификаты

HTTPS является обязательным требованием для современных сайтов. Google использует HTTPS как сигнал ранжирования, а браузеры помечают HTTP-сайты как «Небезопасные». Правильная настройка SSL-сертификата критически важна для безопасности и SEO.

Типы SSL-сертификатов

Тип	Валидация	Стоимость	Когда использовать
Domain Validation (DV)	Подтверждение домена	Бесплатно — \$10/год	Блоги, личные сайты
Organization Validation (OV)	Проверка организации	\$50-200/год	Корпоративные сайты
Extended Validation (EV)	Полная проверка	\$200-1000/год	Банки, крупные магазины

Настройка SSL

```
# Nginx: редирект HTTP → HTTPS
server {
    listen 80;
    server_name example.com www.example.com;
    return 301 https://$server_name$request_uri;
}

server {
    listen 443 ssl http2;
    server_name example.com;

    ssl_certificate /etc/letsencrypt/live/example.com/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/example.com/privkey.pem;
}
```

Чек-лист SSL

- ☐ SSL-сертификат установлен и действителен
- ☐ Все страницы доступны по HTTPS
- ☐ Настроен 301-редирект HTTP → HTTPS
- ☐ Нет mixed content (HTTP-ресурсов на HTTPS-страницах)
- ☐ Сертификат установлен на правильный домен
- ☐ Сертификат не истекает в ближайшие 30 дней
- ☐ Настроен автоматический обновление (Let's Encrypt)

6.2 Mixed content

Mixed content — это ситуация, когда HTTPS-страница загружает ресурсы (скрипты, стили, изображения) по незащищённому HTTP. Браузеры блокируют или предупреждают о таком контенте, что может сломать функционал страницы.

Типы mixed content

Тип	Описание	Поведение браузера
Active (строгий)	JS, CSS, iframe, XHR	Блокируется полностью
Passive (упрощённый)	img, video, audio	Загружается с предупреждением

Поиск и исправление mixed content

1 Сканирование

Используйте Screaming Frog или curl для поиска HTTP-ресурсов на HTTPS-страницах. Поищите в коде шаблона вхождения `http://` без перенаправления на `https://`.

2 Исправление

Замените все `http://` на `https://` или используйте относительные URL (`//example.com/resource`). Проверьте сторонние скрипты и виджеты.

6.3 HSTS и security headers

HTTP Strict Transport Security (HSTS) заставляет браузер всегда использовать HTTPS для данного домена. Security headers дополнительно защищают сайт от различных атак.

Заголовки безопасности

Заголовок	Назначение	Рекомендуемое значение
Strict-Transport-Security	Принудительный HTTPS	<code>max-age=63072000;</code> <code>includeSubDomains; preload</code>
X-Content-Type-Options	Защита от MIME-sniffing	<code>nosniff</code>
X-Frame-Options	Защита от clickjacking	<code>SAMEORIGIN</code>
X-XSS-Protection	Защита от XSS (устаревший)	<code>0</code> (рекомендуется отключить)
Content-Security-Policy	Контроль ресурсов	Индивидуальная настройка
Referrer-Policy	Контроль referrer	<code>strict-origin-when-cross-origin</code>
Permissions-Policy	Контроль API браузера	<code>camera=(), microphone=()</code>

```
# Nginx: заголовки безопасности
add_header Strict-Transport-Security
    "max-age=63072000; includeSubDomains; preload" always;
add_header X-Content-Type-Options "nosniff" always;
add_header X-Frame-Options "SAMEORIGIN" always;
add_header Referrer-Policy "strict-origin-when-cross-origin" always;
```

```
add_header Permissions-Policy "camera=(), microphone=(), geolocation=()"
always;
```

6.4 Чек-лист безопасности

Общий чек-лист безопасности

- ☐ HTTPS настроен для всех страниц
- ☐ 301-редирект HTTP → HTTPS работает
- ☐ Нет mixed content
- ☐ HSTS включён с max-age ≥ 1 года
- ☐ Все security headers настроены
- ☐ CSP policy настроена
- ☐ Сертификат автоматически обновляется
- ☐ Нет уязвимостей в сторонних библиотеках
- ☐ Формы используют CSRF-токены
- ☐ Пароли хэшируются (bcrypt/argon2)

Тестирование

Используйте инструмент securityheaders.com для проверки заголовков безопасности. Целевой уровень — А или А+. Для проверки SSL используйте ssllabs.com/ssltest.

Кейс: Влияние HTTPS на трафик

Интернет-магазин после перехода на HTTPS и настройки всех security headers увеличил органический трафик на 12% за 3 месяца. Особенно вырос трафик с мобильных устройств, где браузеры агрессивнее предупреждают о небезопасных сайтах.

Результат: +12% органического трафика

ГЛАВА 7

Международная SEO

7.1 Hreflang теги

Hreflang теги указывают поисковым системам языковую и региональную принадлежность страниц. Они критически важны для многоязычных и мультирегиональных сайтов, чтобы показывать пользователям правильную версию контента.

Формат hreflang

```
# В HTML (в head)
<link rel="alternate" hreflang="ru" href="https://example.com/ru/" />
<link rel="alternate" hreflang="en" href="https://example.com/en/" />
<link rel="alternate" hreflang="x-default" href="https://example.com/" />

# В HTTP-заголовках
Link: <https://example.com/ru/>; rel="alternate"; hreflang="ru"
Link: <https://example.com/en/>; rel="alternate"; hreflang="en"
```

Правила hreflang

- Все страницы должны ссылаться друг на друга (двусторонние ссылки)
- Используйте правильные коды языков (ISO 639-1) и регионов (ISO 3166-1)
- x-default указывает на страницу для неопределённых языков/регионов
- Для одной страны с одним языком всё равно нужен hreflang

Распространённые ошибки hreflang

Ошибка	Последствие	Решение
Односторонние ссылки	Игнорируется	Добавить обратные ссылки
Неправильные коды языков	Игнорируется	Использовать ISO стандарты
Ссылки на 404 страницы	Игнорируется	Убедиться в доступности URL
Несколько hreflang для одного языка	Конфликт	Оставить один тег
Отсутствие x-default	Нет фолбэка	Добавить x-default

7.2 Геотаргетинг

Геотаргетинг позволяет указать поисковым системам, для какого географического региона предназначен тот или иной сайт или страница.

Методы геотаргетинга

- **Код страны в домене** — example.de, example.co.uk (самый сильный сигнал)
- **Поддомен с кодом языка** — de.example.com
- **Папка с кодом языка** — example.com/de/
- **Отдельные домены** — example.de, example.ru (самый дорогой вариант)

1 Выбор структуры URL

Для большинства сайтов оптимальным является использование папок с кодами языков (example.com/ru/, example.com/en/). Это проще в поддержке и сохраняет авторитет домена.

2 Настройка в Search Console

Укажите целевую страну для каждого свойства в Google Search Console. Это поможет Google правильно определить целевую аудиторию.

7.3 Языковые метки

Помимо hreflang, существует несколько способов указать язык страницы поисковым системам и браузерам.

Языковые метки

Метка	Где	Назначение
lang атрибут HTML	<html lang="ru">	Язык контента страницы
Content-Language	HTTP-заголовок	Язык и регион

Метка	Где	Назначение
hreflang	Link тег / HTTP	Языковые альтернативы
meta robots	HTML	Инструкции для роботов

Чек-лист языковых меток

- ☐ Атрибут lang указан на теге html
- ☐ Язык контента соответствует атрибуту lang
- ☐ Content-Language заголовок настроен (при необходимости)
- ☐ Все языковые версии связаны hreflang тегам
- ☐ Коды языков соответствуют ISO стандартам

7.4 Мультирегиональные стратегии

Для сайтов, ориентированных на несколько регионов, необходимо учитывать не только языковые, но и культурные особенности каждой аудитории.

Факторы мультирегиональной SEO

- **Локализация контента** — не только перевод, но адаптация под местную культуру
- **Валюта и цены** — отображение в местной валюте
- **Доставка и оплата** — информация о местных условиях
- **Контактная информация** — местные телефоны и адреса
- **Социальные сигналы** — активность в местных социальных сетях

Чек-лист мультирегиональной оптимизации

- ☐ Контент локализован (не только переведён)
- ☐ Цены отображаются в местной валюте
- ☐ Настроены местные способы оплаты

- ☐ Указана местная контактная информация
- ☐ Настроены местные методы доставки
- ☐ Созданы ссылки из местных источников
- ☐ Учтены культурные особенности дизайна
- ☐ Проведена SEO-оптимизация для местных поисковых систем

Совет

Для крупных рынков (Китай, Япония, Корея) может потребоваться работа с местными поисковыми системами (Baidu, Yahoo Japan, Naver), которые имеют свои особенности индексации и ранжирования.

Кейс: Международная экспансия

Интернет-магазин одежды после правильной настройки hreflang и локализации контента увеличил трафик из-за рубежа на 156% за 6 месяцев. Ключевым фактором стал переход от машинного перевода к профессиональной локализации с учётом культурных особенностей.

Результат: +156% международного трафика

ГЛАВА 8

Финальный чек-лист

8.1 Критические проблемы (Приоритет: Немедленно)

Эти проблемы должны быть исправлены немедленно, так как они напрямую блокируют индексацию или значительно ухудшают ранжирование.

Индексация и доступность

- ☐ robots.txt не блокирует важные разделы сайта
- ☐ Нет страниц с ошибкой 5xx (внутренняя ошибка сервера)
- ☐ Нет циклических редиректов
- ☐ Нет «soft 404» страниц (404, возвращающих код 200)
- ☐ Доступны все ресурсы, необходимые для рендеринга (CSS, JS)
- ☐ Нет блокировки important pages в robots.txt
- ☐ XML Sitemap доступен и содержит только URL с кодом 200
- ☐ Canonical теги не создают цепочек

Безопасность

- ☐ SSL-сертификат установлен и действителен
- ☐ Настроен редирект HTTP → HTTPS
- ☐ Нет mixed content (HTTP ресурсов на HTTPS страницах)

8.2 Важные проблемы (Приоритет: 1-2 недели)

Эти проблемы значительно влияют на производительность и пользовательский опыт, но не блокируют индексацию.

Core Web Vitals

- ☐ LCP < 2.5 секунд (75% сессий)

- ☐ INP < 200 мс (75% сессий)
- ☐ CLS < 0.1 (75% сессий)
- ☐ TTFB < 800 мс
- ☐ FCP < 1.8 секунд

Мобильная оптимизация

- ☐ Viewport meta tag присутствует
- ☐ Нет горизонтальной прокрутки
- ☐ Размер шрифта ≥ 16 px
- ☐ Интерактивные элементы $\geq 48 \times 48$ px
- ☐ Контент не скрыт за всплывающими окнами

Структурированные данные

- ☐ Основные страницы имеют JSON-LD разметку
- ☐ Разметка проходит валидацию без ошибок
- ☐ Данные в разметке актуальны
- ☐ Для продуктов: name, image, offers, brand
- ☐ Для статей: headline, author, datePublished

8.3 Рекомендации по оптимизации (Приоритет: 1 месяц)

Эти рекомендации помогут улучшить общие показатели сайта и повысить его видимость в поисковых системах.

Контент и структура

- ☐ Для каждой страницы заданы уникальные title и description
- ☐ Заголовки H1-H6 используются иерархически

- ☐ Все изображения имеют alt-атрибуты
- ☐ Внутренние ссылки используют осмысленный anchor text
- ☐ Нет страниц с thin content (менее 300 слов)
- ☐ Хлебные крошки (BreadcrumbList) присутствуют

Производительность

- ☐ Изображения оптимизированы (WebP формат)
- ☐ Реализована lazy loading для изображений
- ☐ JavaScript загружается асинхронно
- ☐ CSS сжат и минимизирован
- ☐ Настроено кэширование статических ресурсов
- ☐ Используется CDN

8.4 Дополнительные пункты (Приоритет: По плану)

Эти пункты являются рекомендациями по улучшению и должны выполняться по мере необходимости.

Продвинутая оптимизация

- ☐ Настроена международная SEO (hreflang)
- ☐ Настроены security headers (HSTS, CSP, X-Frame-Options)
- ☐ Проведён анализ логов сервера
- ☐ Настроен мониторинг Core Web Vitals
- ☐ Реализован AMP (при необходимости)
- ☐ Настроен автоматический аудит (CI/CD)
- ☐ Создан шаблон отчёта для регулярного аудита

- ☐ Настроен алертинг на критические изменения

Документация и процессы

- ☐ Создана инструкция по добавлению нового контента
- ☐ Настроен процесс проверки перед публикацией
- ☐ Создан чек-лист для контент-менеджеров
- ☐ Настроен регулярный повтор аудита (ежеквартально)
- ☐ Документированы все технические решения

Итого

Данный чек-лист содержит 120+ пунктов для полного технического аудита. Регулярное использование этого чек-листа поможет поддерживать техническое здоровье сайта и обеспечивать стабильную видимость в поисковых системах.

Сводная таблица приоритетов

Категория	Критично	Важно	Средне	Рекомендация
Индексация	8	5	3	2
Core Web Vitals	3	5	4	3
Мобильная оптимизация	2	4	3	2
Структурированные данные	1	3	4	3
Безопасность	3	2	3	2
Международная SEO	0	2	3	3
Итого	17	21	20	15

После завершения аудита рекомендуется:

1. Исправить все критические проблемы немедленно
2. Составить план исправления важных проблем на 1-2 недели
3. Запланировать среднесрочные улучшения на месяц
4. Добавить рекомендации в roadmap развития сайта
5. Настроить регулярный повтор аудита (ежеквартально)

Технический SEO аудит

120+ пунктов для полного технического аудита вашего сайта. Используйте этот чек-лист регулярно для поддержания технического здоровья сайта.

- ✓ 8 глав с подробными инструкциями
- ✓ 120+ пунктов чек-листа
- ✓ Готовые шаблоны и примеры кода
- ✓ Система приоритизации проблем
- ✓ Актуально для 2026 года
- ✓ Примеры из реальных проектов

RankBoost