

## Аудит сайта (сайт, нагрузка, импорт из 1С)

### 1. Общий анализ сайта (качество программирования, следование стандартам), нагрузка и производительность

#### 1.1. Тестирование производительности штатным монитором

Инструментальное исследование на сервере Заказчика показывает что многие страницы открываются по 17-38 секунд

Средняя производительность 0.09 (замер 3600 секунд, 102 хитов) 18.01.2013 12:44:23

20 самых нагружающих страниц	Ошибки разработки <sup>2</sup>	Нагрузка	Количество хитов	Среднее время (сек.)
<a href="/catalog/index.php">/catalog/index.php</a>		96.88%	62	17.0591
<a href="/info/news/index.php">/info/news/index.php</a>		1.86%	2	10.1559
<a href="/bitrix/urlrewrite.php">/bitrix/urlrewrite.php</a>		1.01%	36	0.3057
<a href="/index.php">/index.php</a>		0.18%	1	1.9130
<a href="/info/special-offers/index.php">/info/special-offers/index.php</a>		0.08%	1	0.8366

[Все страницы](#)

Средняя производительность 0.09 (замер 1800 секунд, 120 хитов) 18.01.2013 16:50:40

20 самых нагружающих страниц	Ошибки разработки <sup>2</sup>	Нагрузка	Количество хитов	Среднее время (сек.)
<a href="/catalog/index.php">/catalog/index.php</a>	1	78.22%	60	17.4317
<a href="/info/news/index.php">/info/news/index.php</a>	1	9.71%	7	18.5457
<a href="/personal/basket/index.php">/personal/basket/index.php</a>	1	3.10%	3	13.8379
<a href="/personal/index.php">/personal/index.php</a>	1	2.93%	1	39.1854
<a href="/bitrix/urlrewrite.php">/bitrix/urlrewrite.php</a>		2.69%	28	1.2856
<a href="/info/special-offers/index.php">/info/special-offers/index.php</a>	1	0.92%	3	4.1050

## 1.2. Нагрузочное тестирование

Практически провести не удалось. При любой нагрузке свыше 10 просмотров в минуту и одновременной работе многих людей сайт работает все медленнее и медленнее и работать становится невозможно, монитор производительности выдает ошибку 502.

Сайт не готов держать никакую реальную нагрузку, несмотря на хорошее оборудование.

## 1.3. Исследование нагружающих компонент и запросов

Если включить отладку “с лица” сайта и отследить время исполнения запросов и их число, получается вот такая картина (при включенном по нашей рекомендации “управляемом кеше”, изначально он был выключен):

### 1. Список товаров

Всего SQL запросов: 254  
Время исполнения запросов: 1.9308 сек.  
Объем кеша: 362 КБ

### 2. Новости

Всего SQL запросов: 31  
Время исполнения запросов: 0.0899 сек.  
Объем кеша: 283 КБ

### 3. Корзина

Всего SQL запросов: 98  
Время исполнения запросов: 0.2046 сек.  
Объем кеша: 313 КБ

Это очень много. Десятки запросов на простые функции и сотни на товары -- неприемлемо много. Для сравнения - аналогичные страницы на недавно сданном нами среднем интернет-магазине [http://\\_\\_\\_\\_.ru/](http://____.ru/) :

### 1. Список товаров

Всего SQL запросов: 8  
Время исполнения запросов: 0.0000 сек.  
Объем кеша: 406 КБ

Лучшая [садовая те](#)

### 2. Новости (спец. предложения в виде новостей)

Всего SQL запросов: 6  
Время исполнения запросов: 0.0000 сек.  
Объем кеша: 264 КБ

Лучшая [садовая те](#)

### 3. Корзина

Всего SQL запросов: 61  
Время исполнения запросов: 0.0000 сек.  
Объем кеша: 269 КБ

Лучшая [садовая те](#)

Описанные выше данные получены в условиях чистого эксперимента: при повторном открытии каждой из страниц и сгенерированном кеше.

Есть острая необходимость в оптимизации количества запросов путем изучения и переписывания компонентов.

## 1.4. Исследование программного кода в проблемных местах

### Краткий вывод

Очень низкое качество кода. Присутствуют ошибки как логического характера, так и не оптимальный код и откровенные ошибки грубого характера. Общей стилистики кода нет, код читается тяжело и требуется большое время на понимание того или иного функционала.

Детали (проанализированы ключевые участки, создающие наибольшие проблемы):

1. На всех страницах сайта идет включение кода, который выполняется на любой странице сайта (включая админку), при необходимости и без нее. Кеширование в данном случае не используется. Этот код понижает оценку производительности битрикса более чем в 10 раз.

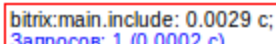
2. На сайте во многих местах нарушен принцип разделения логики и представления данных.

Например файл:

- /inc/includes/functions.php, строка 20, функция can\_control. В функции текст, который в итоге выводится на экран уже задан в ней самой. В целом функция создана непонятно зачем, путает разработчика, который будет поддерживать код. Некритично, но показательно.
- Во включаемой области с выводом бонусов пользователя подключается файл /personal/sect\_inc.php



У вас бонусов: 0



bitrix:main.include: 0.0029 c;  
Запросов: 1 (0 000? c)

При этом в данной области html код вперемешку с php кодом (любой редактор, имеющий права на редактирование, может случайно испортить этот код). Также в этом блоке не используется кеширование.

- В файле: /bitrix/templates/.default/components/bitrix/catalog/goods/filter.php содержится код, отвечающий за фильтр. В шаблоне вперемешку запросы на API битрикса и html. Сам файл не кешируется в случае открытия корня каталога. Все вперемешку и может впоследствии перестать работать корректно при изменении другим разработчиком.

3. В большей части сайта либо не используется кеширование, либо оно используется не корректно.

- На главной странице в блоке спецпредложения, в шаблоне имеются запросы в файле /bitrix/templates/.default/components/bitrix/news.list/index\_offers/template.php. Также не установлена галочка “кешировать при установленном фильтре”, что приводит к большой нагрузке и большому количеству запросов. Фильтр задан вручную и не меняется, а без этой галочки и без того неоптимальный компонент работает еще без кеша и отправляет на 4 элемента 10 запросов.
- По сайту не включено кеширование меню, что фиксируется монитором качества 1С-Битрикс.
- Редактирование личных данных: /personal/ Компонент iarga:main.profile

Не используется кеширование списка городов (отдельный инфоблок) и списка “откуда вы о нас узнали” (отдельный инфоблок).

4. В разных местах сайта обнаружены примитивные ошибки. Так же во многих местах код написан не оптимально.

- Ошибки в коде подключаемого файла `/bitrix/php_interface/init.php`
  - a. в функции `OnSalePayOrder` в строке 32 существует вывод на экран: `“print $bonuses;”` которого там быть в принципе не должно: операции вывода на экран не проводятся из функций, это инструмент отладки, забытый программистом. Некритично, но показательно.
  - b. функция `OnAfterIBlockElementAddHandler` строка 42-43. Закомментированный код, притом что сама функция есть и вызывается. Сама функция и событие, которое ее инициализирует потребляет ресурсы без надобности. Некритично, но показательно.
  - c. функция `OnAfterUserUpdate`. Смысл функции в том, что она в случае изменения данных пользователя по пути `/personal/` удаляет из списка его групп группу с ID 9 (если она присутствует). Данное решение является не оптимальным.  
Оптимальным было бы в компоненте редактирования профиля по пути `/personal/` код разместить. В текущей реализации этот код вызывается при любом изменении в профиле пользователя и тратит ресурсы.
- Верхнее меню генерирует 2-4 запроса (это компонент `catalog.section.list` доработанный программистом), и время этих запросов колеблется от 100 до 200 секунд.  
Причина: В файле `/inc/parts/header.php`, который подключается на всех шаблонах используется компонент `iarga:catalog.section.list`, у которого включен параметр `COUNT_ELEMENTS`, который отвечает за подсчет количества элементов в разделах. Так как каталог большой, то данный запрос существенно тормозит генерацию страницы.  
**200 секунд на меню** -- это категорически неприемлемо даже при полном отсутствии кеша.  
Критично.
- Во всех шаблонах сайта вверху в файле `/inc/parts/head.php` Там в коде написан `html`, объявление кодировки страницы и подключение `'keywords'` и `'description'`. Но далее для администратора и страницы поиска (не понятно почему) используется код, который опять подключает вывод метатегов `keywords` и `description` и опять объявляет кодировку (причем в первом случае она прописана в коде, во втором она зависит от настроек сайта). В результате имеем не валидный `html`, возможные проблемы с кодировкой, возможные проблемы с поисковыми системами (неизвестно, как они отреагируют на дублирование `'keywords'` и `'description'`).  
Критично.
- “мы в соц. сетях” не имеет возможности редактирования для редактора сайта без входа в сложную правку шаблона страницы. Данный код размещается на странице `/inc/parts/footer.php` и может быть только отредактирован программистом в указанном файле (редактор без достаточных знаний рискует при редактировании испортить существующий код).

Некритично, но показательно и потенциально опасно.

- В верхней части шаблона сайта версии для печати в файле `/bitrix/templates/print/header.php` руками установлена кодировка `windows-1251`, что не соответствует действительности (сайт отдается в UTF8).

Некритично, но показательно и потенциально опасно.

- На странице сайта `/info/shop/contact_information.php` присутствует форма обратной связи. Данная форма не реализована с помощью компонент, как принято (в битриксе для этого существует готовый компонент). Исполняемый код напрямую подключается на странице.

Неоптимально.

- Страница отзывов и предложений: `/info/shop/feedback_and_suggestions.php`
  - а. в случае успешной отправки отзыва никакого сообщения об этом не выводится. Форма просто пропадает. Это сбивает с толка пользователя.  
Неудобно.
  - б. форма добавления отзыва находится в кешируемой области, а не вынесена в отдельный компонент (нарушение принятых в кодировании под 1С-Битрикс правил).  
Это приводит к тому, что капча не обновляется при обновлении страницы до сброса кеша.
- Если открыть каталог через “хлебные крошки”, то фильтр не устанавливает значения по умолчанию.  
Неудобно

5. В каталоге тоже плохой неоптимальный код, который в сочетании с большим числом товаров создает проблемы производительности (товаров более 40 тыс и свойств более 4000).

Также штатные механизмы 1С-Битрикс, его обычные функции дублируются, что приводит к двукратной нагрузке при открытии страниц. Также, в шаблонах битрикса используются запросы. Это грубое нарушение логики битрикса и принципов и правил программирования.

Далее подробно:

- В списке товаров, в файле, который отвечает за вывод карточки каждого товара в шаблоне делается запрос на свойства, на оптимальную цену и на скидки. Данное решение является не корректным и при сбросе кеша порождает большое количество избыточных запросов к базе данных.  
Файл: `/inc/parts/good.php`
- Детальная страница элемента:
  - Файл  
`/bitrix/templates/.default/components/bitrix/catalog/goods/creat/catalog.element/.default/component_epilog.php`  
Делаются не кешируемые запросы, так же дублируются штатные возможности битрикса по заданию метаданных страницы.
  - Файл  
`/bitrix/templates/.default/components/bitrix/catalog/goods/creat/catalog.element/.default/result_modi`

fier.php

Область кешируемая, но присутствует большой объем не оптимального кода, который генерирует много запросов. Вторично получается информация, которая уже получена была компонентом.

- **Файл**

/bitrix/templates/.default/components/bitrix/catalog/goods/creat/catalog.element/.default/template.php

р Шаблон детальной страницы элемента. Присутствуют API запросы в самом шаблоне, что является не допустимым.

6. Также на сайте включен ряд модулей, которые не используются на сайте и их отключение позволит оптимизировать нагрузку (до 30% прироста скорости):

- Веб-аналитика (существенно снижает нагрузку на сайт)
- Веб-мессенджер
- Веб-сервисы
- Веб-формы
- Документооборот
- Календарь событий
- Подписка, рассылки
- Социальные сервисы
- Универсальные списки
- Форум

### 1.5. Странный файл bx\_1c\_import.php и уязвимости

В корне сайта лежит доступный всем файл для импорта из 1С /bx\_1c\_import.php.

Этот файл -- не часть системы 1С-Битрикс. Монитор безопасности сообщает что в нем есть найденные путем статического анализа кода уязвимости (Проверена безопасность кода (статический анализ уязвимостей) (QSEC0080)) - 37 замечаний.

Вероятно (судя по названию файла и комментариям внутри) этот файл размещен там в рамках общения с техподдержкой 1С-Битрикс.

Файл надо удалить.

### 1.6. Вывод по анализу программного кода

Программная часть сайта реализована плохо, с большим количеством ошибок, неточностей и “неряшливостей” кода (ошибок, которые не являются важными и критичными, но снижают конечное качество продукта).

Ошибок на сайте такое количество, что конкретного места, которое бы приводило к медленной работе сайта, просто нет. Все ошибки в той или иной мере накладываются друг на друга.

Начать следует с каталога товаров.

В текущей реализации качества сайта низкое и по-хорошему сайт необходимо переписать с нуля.

В рамках данной работы проводился аудит только кода, верстка и корректность работы пользовательских интерфейсов не проверялась. Но даже в данном случае были обнаружены ошибки верстки и ошибки логики работы сайта. При этом сайт не является особо сложным, а представляет собой средний интернет-магазин.

Наши проекты для \_\_\_ и \_\_\_ по сложности превосходят данный сайт в несколько раз.

Возможно несколько вариантов решения текущей проблемы с сайтом:

1. Заставить этот код работать быстрее.

Только оптимизация кода, переписывание некоторых мест, оптимальное использование кеширования. В данном случае нам не нужно знать ничего о сайте, как он должен работать, какая логика заложена. Наша задача провести оптимизацию кода таким образом, чтобы работало как и ранее по логике, а по скорости быстрее. При этом данный сайт будет так же тяжело поддерживать на уровне кода, как и сейчас (но будет решена проблема производительности сайта). Сложность такой реализации 1-2 рабочие недели, т. е. 40-80 часов рабочего времени.

2. Переписывание всего кода, оставляя верстку и логику сайта как сейчас, устраняя откровенные ошибки (не работа фильтра в корне сайта, не работа капчи, не вывод сообщения об отправке и ошибок). При этом нам нужно будет ТЗ на сайт и согласование с заказчиком ряда вопросов. При этом никакие претензии к верстке не принимаются и по логике только ошибки, которые не соответствуют ТЗ.

Сложность работы: 160-200 часов. Перед работой нужно ознакомиться с ТЗ (вдруг там есть вещи, которые вообще не реализованы).

В результате код будет быстро работать, и его станет можно развивать и поддерживать.

Варианта правки верстки нет: только переписывание с нуля (ну или почасовая работа в рамках тех. поддержки).

3. Полная переделка сайта.

Все будет хорошо, но достаточно долго и дорого.

Мы готовы сделать предложение.

## 2. Импорт из 1С

### 2.1. Статистика по обмену

Если в базе, к которой нам был предоставлен доступ был верно указан адрес страницы через которую идет обмен сайта и 1С (`/bitrix/admin/1c_exchange.php`) - то это означает что на стороне сайта используется 100% **стандартный модуль обмена**.

Этот вывод получается из следующих шагов:

1. Указанный выше файл ссылается на стандартный компонент в ядре битрикса

## 2. Монитор качества говорит, что в этом месте ядро не модифицировалось

Замер временных затрат в ходе выгрузки изменений по 3000 товаров показал следующие результаты:

Этап	Время в минутах
Предварительная обработка данных на стороне 1С (поиск того что нужно выгружать)	1.5
Выгрузка из 1С на диск информации о товарах	1.5
Выгрузка из 1С на диск информации о ценах	менее 1
Передача файлов выгрузки на сайт	1
Загрузка сайтом во временные таблицы	4
Загрузка сайтом в инфоблок	28

**Структура временных расходов при обмене с 1С**



Учитывая объем выгрузки (изменения по 3000 товаров) можно сделать следующие выводы:

- Модуль обмена на стороне 1С вполне быстро справляется со своей задачей. Приблизительно такую же производительность мы видим в своих проектах.
- Время загрузки данных во временные таблицы на сайте приблизительно в 2-4 раза превышает аналогичное время на наших проектах.
- Время загрузки данных из временных таблиц в основное хранилище (инфоблок) значительно превышает аналогичное время на наших проектах - в 5-8 раз.

**Вывод: проблема долгой выгрузки на стороне сайта.**



## Основные проблемы, снижающие скорость обмена

### 1. Низкая производительность сайта

Сайт загружает данные (как во временные таблицы, так и в инфоблок) порциями. Чтобы загрузить очередную порцию 1С запрашивает одну из страниц сайта (аналогично тому как Вы открываете сайт в браузере). В результате все, что разработчики выполняют при открытии страницы (/bitrix/php\_interface/init.php) выполняется и при загрузке каждой порции данных из 1С.

Вследствие крайней неоптимальности кода эта операция каждый раз занимает очень много времени. Фактически каждая позиция загружается 5-10 секунд. Нормальным является показатель не более 1 секунды на позицию.

### 2. Большое количество характеристик товара

Из 1С на сайт выгружено более 4000 характеристик в сумме (это аналогично 40 типам товаров со 100 уникальными характеристиками в 1С). Это в действительности огромная цифра и есть сомнения что такое количество характеристик оправдано. Их стоит создавать чтобы либо использовать в фильтрах, либо выводить структурировано в сравнении товаров.

### Вывод:

- **необходимо увеличить производительность сайта (начать с этого, может и будет достаточно)**
- **уменьшить суммарное количество характеристик в 1С (предельно целесообразным видится число до 500) (второе по приоритету).**
- **Дополнительные шаги**

Эти шаги могут могут незначительно (в сумме не более 10%) увеличить скорость обмена с 1С:

1. Использовать для хранения каталога механизм "Инфоблоки 2.0". Учитывая текущее количество характеристик это нельзя сделать. Кроме того есть определенные проблемы при работе этого механизма при используемом Вами типе таблиц базы данных (InnoDB). Самостоятельно пытаться выполнить эту рекомендацию не советуем.
2. Провести оптимизацию базы данных
3. Подобрать более оптимальные настройки обмена
  - Отключить флажки "Использовать zip-сжатие" на всех вкладках. Эта функция на большинстве хостингов работает с ошибками. Включать ее имеет смысл если у Вас очень медленная скорость передачи файлов по сети Интернет.
  - Включить флажок "Использовать контрольные суммы элементов для оптимизации обновления каталога". Это может уменьшить время загрузки данных на сайт.
  - На закладке "Каталог" увеличить значение "Интервал одного шага в секундах (0 - выполнять загрузку за один шаг)" с 10 секунд до максимально возможного (обычно 20-30 секунд). Это немного ускорит обмен. При установке данного значения в 25 и более секунд нужно убедиться что сайт работает в режиме Apache.

- На закладке “Экспорт каталога” установить значение “Количество элементов выгружаемых за один шаг (0 - выполнять выгрузку за один шаг)” вместо 1 - в несколько раз больше. Как минимум 10. В будущем увеличивать на 10 и наблюдать за тем, что обмен стал выдавать ошибки.

### 3. Выводы

В ходе аудита нами не было внесено никаких изменений в работе сайта, все исследования проводились на отдельной копии.

Предполагаем что наведение порядка на сайте способно ускорить его работу от нескольких раз до нескольких десятков раз. Предполагаем что при посещаемости сайта до 1000 человек в сутки его выдержит виртуальный хостинг Etemo компании Timeweb стоимостью 8415 рублей в год.

В наших магазинах мы сталкиваемся с нехваткой ресурсов этого тарифа от 2000 человек в сутки.

Руководитель группы разработки	_____	Покоев С.В.
Руководитель отдела управления проектами	_____	Шкарупа А.Э.
Ген.директор		
ООО “Интернет-агентство ИНТЕРВОЛГА”	_____	Овчинников С.А.