

Аудит качества программного кода сайта

1. Проверка сайта штатными средствами CMS 1С-Битрикс

1.1. Проверка сайта

Файлы сайта

Есть файлы, не доступные для чтения или записи:

- /home/.../ayment/.SistemaPag
- /home/.../ayment/.SistemaPag/index.php
- /home/.../ayment/.SistemaPag/Cliente-2013009158/Fatura05-2013/FaturaVencimento05-2013.zip
- /home/.../ayment/.SistemaPag/Cliente-2013009158/Boleto07-2013/BoletoVencimento07-2013.zip

Ядро проекта **не модифицировалось**.

Монитор качества показывает, что в main модуле модифицированы 3 файла:

- /home/.../bitrix/modules/main/admin/update_system.php
- /home/.../bitrix/modules/main/classes/general/update_class.php
- /home/.../bitrix/modules/main/classes/general/update_log.php

но это связано с отсутствием последних обновлений.

Производительность конфигурации

Конфигурация используемого сервера достаточна для работы интернет-магазина, все проверки пройдены.

В будущем при росте нагрузки или добавлении функций может потребоваться повысить скорость работы процессора, число ядер, скорость работы дисковой системы.

Конфигурация (47.72) Битрикс (не оптимально) Разработка (0.90) Масштабируемость

Производительность конфигурации на 31.10.2013 08:58:45 составляет 47.72

Подсистема	Оценка	Эталон	Примечание
Конфигурация	47.72	30	
Среднее время отклика	0.0210	0.0330	секунд
Процессор (CPU)	8.3	9.0	миллионов операций в секунду
Файловая система	7 487.9	10 000	файловых операций в секунду
Почтовая система	0.0071	0.0100	время отправки одного письма (в секундах)
Время старта сессии	0.0004	0.0002	секунд
Конфигурация PHP	оптимально	оптимально	рекомендации
База данных MySQL (запись)	14 442	5 600	количество запросов на запись в секунду
База данных MySQL (чтение)	17 123	7 800	количество запросов на чтение в секунду
База данных MySQL (изменение)	9 711	5 800	количество запросов на изменение в секунду

[Тестировать конфигурацию](#)

Конфигурация веб-сервера и ПО

Настройка	Значение	Рекомендации
Автокеширование компонентов	Включено	
HTML кеш	Выключен	
Настройки модуля поиска	Быстрый морфологический поиск выключен	Включите морфологический поиск и опцию быстрого поиска
Хранение кеша	Файлы	Возможные типы хранения: <ul style="list-style-type: none"> • Файлы • memcached • eAccelerator • APC • XCache Инструкция по настройке.
Управляемый кеш	Включен	
Закодированные модули	Не найдены	
Оптимизация и анализ таблиц базы данных	Необходимо выполнить	Оптимизировать
Модуль компрессии	Установлен	

Рекомендуется выполнить советы 1С-Битрикс. Эти рекомендации не являются критичными, но сделать

это желательно.

Не соответствует конфигурация PHP, а именно: значение `max_input_vars` должно быть не ниже 3000. Текущее значение: 1000. Могут возникнуть проблемы в работе с инфоблоками, имеющими большое количество свойств. Ограничение ниже рекомендуемого предела приводит к тому, что при сохранении формы с большим количеством полей произойдет не полное сохранение формы или ошибочная обработка.

1.2. Тестирование скорости работы

Тестирование производительности встроенными инструментами платформы 1С-Битрикс дает хорошие показатели работы сайта.

Среднее время генерации страниц 0.4 секунды при показах от 10 до 100 страниц в секунду.

Идеальным считается время генерации менее 0.1 с, плохими - более 1 с.

Учитывая написанное далее, можно сказать, что отличное железо вытягивает плохой код на нормальные показатели.

1.3 Исследование используемых шаблонов, компонентов, качества кода, примененных решений

Шаблон

В шаблоне сайта большое количество прямых запросов на API. с нарушением правил программирования, с жестко “защитыми” параметрами. Пример: шаблон главной страницы содержит пользовательскую функцию, внутри которой написан код на API.

Это неправильно с точки зрения идеологии и неоптимально по нагрузке. Подобные решения применены повсеместно.

Компоненты

Количество запросов создаваемых используемыми компонентами, а именно `bitrix:catalog.section`, `bitrix:catalog.element`, `bitrix:`, очень велико:

- раздел `/catalog/___i/`

```
bitrix:catalog.section: 0.5279 с; Запросов: 910 (0.1661 с); кеш: 6 КБ
```

- другие разделы каталога `/catalog/#SECTION_ID#/`

Меню разделов

Меню разделов реализовано некорректно. Это выполнено в виде включаемой области, расположенной в левом блоке. На страницах разделов каталога, и на других страницах сайта данный элемент вполне корректно работает:

```
bitrix:catalog.section: 1.3679 с; Запросов: 85 (1.3061 с); кеш: 27 КБ
```

```
/include_areas/left_menu.php:  
0.0187 с; Запросов: 4 (0.002  
с); кеш: 4 КБ
```

но на страницах товаров создаёт большое кол-во запросов:

```
/include_areas/left_menu.php:  
0.2102 с; Запросов: 138 (0.1327  
с); кеш: 4 КБ
```

Вышеупомянутые проблемы, связанные с большим количеством запросов, вызваны некорректной реализацией шаблонов этих компонентов. В коде шаблонов можно наблюдать множественное использование API, что является неуместным грубым нарушением.

Каталог

Не корректно реализован раздел каталога: http://___/catalog/ как с точки зрения кода, так с точки зрения пользователя.

В данном разделе уместно использовать комплексный компонент. Вместо этого тут используется большое множество кода, написано не универсальным образом, не использующего кеширование и с нарушением правил программирования на Битриксе. Поддержка и модификация данного кода не представляется оправданной по технико-экономическим показателям, так как время на любую правку будет значительно большим, чем при правильной реализации.

Так же невозможность корректного тестирования такого кода (т. е. постоянно возникающие ошибки) и проблематичность понимания того, что тот или иной код делает.

Подробнее:

В файле `/catalog/index.php` используется большое количество не оптимального кода, который не кешируется, а так же происходит избыточное получение данных. Так же в коде прописано множество параметров, т. е. штатная возможность 1С-Битрикс, где можно с помощью настроек изменить вывод данных, их представление, тип и т. п. - не используется. Так же программисту, который будет вносить правки и изменения придется искать множество мест, чтобы изменить те или иные параметры.

Страница раздела каталога (списка товаров)

В разделе http://___/catalog/___/ происходит выборка с огромным количеством запросов

```
bitrix:catalog.section: 0.5279 с; Запросов: 910 (0.1661 с); кеш: 6 КБ
```

Кеш в данном случае не работает.

Нагрузка создается не оптимальным кодом в файле `/bitrix/templates/.default/components/bitrix/catalog.section/1_row/result_modifier.php`

Так же, в шаблоне `/bitrix/templates/.default/components/bitrix/catalog.section/1_row/template.php` используются вызовы на API, что является грубым нарушением правил программирования на Битриксе.

В итоге одни и те же данные получаются многократно.

На других страницах каталога используется другой шаблон `/bitrix/templates/.default/components/bitrix/catalog.section/3_rows`. К нему аналогичные замечания. Ввиду меньшего объема данных, количество запросов меньше, но все равно в недопустимом количестве.

Присутствуют мелкие огрехи: в шаблонах не проверяется наличие на вывод картинки или нет.

Подобные же проблемы на детальной странице каталога.

Верхнее меню

Верхнее меню реализовано недопустимым способом, исключаяющим его правку не программистом.

Поиск

http://___/search/index.php?q=1&x=-903&y=-479&how=r

Не оптимальное получение данных в цикле. Поэтому большое количество запросов. Заголовок страницы не устанавливается.

Новинки

http://___/novels/

Нет такого шаблона.

Главная

Новинки

Can not find 'customcatalog' template with page "

Аналогичная ошибка, распродажи: http://___/sales/, “Рекомендует” http://___/recommends/, “Ожидаем в продаже”: http://___/expectations/,

Блок “Бесплатные схемы”

Контент жестко “зашит” в шаблоне и не предполагается его редактирование редактором.

Разделение логики и представлений

В целом, на сайте полностью нарушена концепция MVC. В представлениях компонентов, а также на многих страницах сайта, даже в шапке и подвале, присутствуют описания функций. Данное решение разработчиков приводит к отсутствию кэширования в компонентах и существенно затрудняет дальнейшее развитие сайта.

Нарушены элементарные принципы программирования на битриксе. Компоненты используются не правильно, либо не используются вообще. Кеширование не работает. Данные получают не оптимальными способами или одни и те же данные многократно.

В коде, при работе с массивами, используются неопределённые константы (т.е. вместо `$myArray['key']\$myArray["key"]\$myArray[$key]` используется конструкция `$myArray[key]`), в дальнейшем, при расширении функционала сайта, это может привести к непредвиденным последствиям, вплоть до отказа работы сайта.

Странности

1. В разделе `/classified/` лежит файл `restoreme.php`, позволяющий сменить пароль любому пользователю, если подобрать соответствующее значение для входящей переменной `'pass'` и передать существующий логин в переменной `'user'` и новый пароль в `'password'`.

2. В разделе `/include_areas/` есть вложенный раздел, название которого содержит кириллические символы, что скорее всего связано с возможностью просматривать родительский раздел из административной панели.

3. В разделе `/include_areas/framework/` лежит файл `___copyright.php`, загружающий (http://___ru/api/links.xml файл загружается, парсится и результаты кэшируются. В блоке "Вёрстка", в этом же файле, вывод результатов отсутствует. Подключается в футере) внешние ссылки с сайта `___`.

4. Непонятно назначение файла `index.php` в разделе `/site_tg/`. Судя по содержимому предназначался для установки собственного решения для битрикса.

В целом по анализу кода можно дать рекомендацию: переписать практически весь функционал (исключая блок и страницу новостей, оформление заказа и корзину) на компоненты, с кешированием и

оптимальным получением данных. Шаблон сайта полностью переделать, нормальным образом формировать хлебные крошки и заголовки страниц.

Это существенно снизит нагрузку на сервер, позволит редактору сайта править тексты, контент и т. п. без привлечения программиста и стоимость дальнейшей поддержки и развития сайта существенно снизит.

1.4 Сканер безопасности

Анализ сайта сканером безопасности показывает 8 угроз, из них по мнению платформы 4 критичны. Не смотря на то, что статический анализ выделяет наиболее критичным проблему с отсутствием проверки получаемых от пользователя значений и предлагает решить это добавлением дополнительных условий, всё же решение является некорректным, т.к. встроенный инструмент безопасности неверно анализирует код и предложенные условия уже есть в коде.

Подробнее в пункте “Анализ программного кода”.

Так же стоит отметить проблему с неверными правами доступа у файлов:

```
Необходимо отобрать лишние права у всех системных пользователей
Последние 5 файлов/директорий:
/payment/.SistemaPag/Cliente-2013009158
/payment/.SistemaPag/Cliente-2013009158/index.php
/payment/.SistemaPag/Cliente-2013009158/Fatura05-2013
/payment/.SistemaPag/Cliente-2013009158/Fatura05-2013/index.php
/payment/.SistemaPag/Cliente-2013009158/Boleto07-2013
```

и Automatic MIME Type Detection для Internet Explorer(по умолчанию в Internet Explorer включен автоматический mime-сниффинг, что может служить источником XSS нападения).

2. Анализ источников сообщенных заказчиком ошибок

2.1 Вывод товара “_____”

При сравнении страницы /catalog/___/117-muline-dmc-khlopkovoe/ и приложенного макета данной страницы, не наблюдаю различий и описанных проблем. Карточка товара раздела “мулине” с действующего сайта: http://___/hijFpRP.jpg

2.2 Ошибка в блоках справа

Данная проблема возникает из-за отсутствия соответствующих шаблонов (в разделах “Новинки”, “Ожидаем в продаже”, “Распродажа” используется шаблон customcatalog) для компонента bitrix:catalog.section. В основном шаблоне сайта(main) имеются представления для данного компонента со схожим названием, но в количестве 4 штук(customcatalog1, customcatalog2, customcatalog3 и customcatalog4).

2.3 Импорт товаров

На сайте используется стандартный импорт CSV, с помощью которого и импортируются новые

товары, согласно приложенной инструкции.

3. Вывод

Сайт реализован программистами, которые, несмотря на явный опыт в разработке сайтов, не знают и не применяют принципы программирования на 1С-Битрикс.

Качество кодирования низкое, множество источников скрытых ошибок.

Варианты действий:

- переписать все некачественные участки кода и развиваться нормально, имея гарантию корректной работы. Это потребует 30-60 часов работы.
Вариант предполагает инвестиции сейчас и предсказуемость потом, низкую стоимость владения, экономию нервов.
- исправлять только откровенно неработающее и наращивать функции.
Вариант предполагает экономию денег вместе с постоянными проблемами в работе и высокой стоимостью владения. Каждая доработка будет обходиться в 1.5-2.5 раза больше чем если все поправить без возможности с нашей стороны дать гарантию, что проблема исправлена окончательно и не проявится в дальнейшем в другом виде..

31 октября 2013 года

Программист, сотрудник службы поддержки проектов

Сергей Березов

Главный специалист

Сергей Покоев

Ген.директор

Степан Овчинников