

# Аудит интернет-магазина \_\_\_ http://\_\_\_/

**Цель аудита:** оценка технического качества кодирования сайта, соблюдения требований к разработке сайтов на 1С-Битрикс.

Основной вопрос, на который должен дать ответ выполняемый аудит: что из разработанного может быть использовано повторно при передаче разработки другой компании? Особо важна возможность сохранения текущей реализации передачи информации из системы 1С в сайт.

**Ход аудита:** специалисты подключаются к сайту как администраторы, запускают стандартные тесты, анализируют программный код, выполняют собственное тестирование и анализ. Никаких изменений программного кода, в том числе для исправления ошибок, не проводится.

## Архитектура каталога. Структура данных. Элементы, свойства, комплекты

Функциональность сайта такова, что подразумевает использование комплектов товаров (товаров, формируемых из других товаров путем сбора их в комплект и назначения им общего названия и цены). С 14 версии система 1С-Битрикс начала поддержку такого типа товаров.

На анализируемом сайте штатные возможности 1С-Битрикс не используются, применено самописное, неуниверсальное и неоптимальное решение. Объяснить это можно тем, что версия 14 вышла только осенью 2013 года.

Работа с комплектами на сайте основана на структуре данных каталога, подразумевающей хранение всех исходных данных и сбор комплектов на сайте программным кодом.

Текущая архитектура каталога, структура данных и свойства решает только часть задач и существенно ограничивает (делает дорогим и неудобным) добавление функций и возможностей.

Причиной выбора именно такой архитектуры может быть специфика учета в 1С (сайт работает не на основе всех возможных торговых предложений, а формирует предложения по параметрам, приходящим из 1С).

Частично текущую реализацию можно было сделать более универсальной за счет реализации комплектов теми средствами которые предоставляет 1С-Битрикс.

Примеры функциональных ограничений:

- При текущей архитектуре каталога невозможно будет вывести в списке товаров корректную цену, т.к. текущая реализация комплектов (как товар со связанными товарами из 1С) не будет поддерживать автоматический расчет цен на основе товаров, входящих в комплект. Точную цену

комплекта можно будет увидеть только в карточке товара. В поиске, в списке товаров вывести цену можно только через т.н. “костыли”, то есть неудобные неоптимальные нетиражируемые решения.

- Скидки в 1С-Битрикс реализованы таким образом, что могут применяться к некоторым товарам комплекта. В таком случае цена комплекта должна уменьшаться, однако при текущей реализации этого не произойдет (нельзя будет отобразить эту цену в списке товаров с учетом скидки).
- При текущей реализации механизма хранения характеристик товаров невозможно будет сделать фильтр в списке товаров по большинству характеристик. Фильтр сможет работать только по тем характеристикам, которые выгружаются именно в “Товар”.
- В корзине будет непросто отображать заказанные комплекты как 1 строчку (как это описано в ТЗ).
- При текущей структуре невозможно организовать контроль остатков в той мере, которая требуется по ТЗ.

## Код

### Каталог ([http://\\_\\_\\_/catalog/](http://___/catalog/))

Используется комплексный компонент каталог (что правильно), но шаблоны входящих в него не комплексных компонентов лежат не в шаблоне комплексного, хотя больше нигде не используются (что странно и затрудняет использование решения). Возникает ощущение что автор кода не знал как правильно из-за недостатка опыта.

Страница каталога и раздела каталога не сделаны до конца и полностью оценены не могут быть.

### Детальная страница каталога ([http://\\_\\_\\_/catalog/\\_\\_\\_/](http://___/catalog/___/)):

Шаблон детальной каталога (`/bitrix/templates/micro___-new/components/bitrix/catalog/catalog_page_new/bitrix/catalog.element/.default/template.php`) содержит объявления функций, запросы к базе через api.

Это грубое нарушение правил системы 1С-Битрикс, которое указывает что программный код API и для работы с базой данных можно писать только в компонентах, в шаблонах можно только выводить подготовленные в компоненте данные через html-код.

Последствием этого решения являются:

- невозможность кеширования указанного компонента
- снижение производительности и необоснованное повышение нагрузки на сервер
- невозможность повторного использования кода

Странное решение выводить информацию об оплате/доставке через показ элемента инфоблока с определенным кодом, а не через включаемые области. Код элементов жестко указан в коде, как и, например, телефон “Позвоните нам”.

Последствием этого решения являются:

- тот факт, что если администратор случайно удалит элемент инфоблока, восстановит его потом, соответствующий кусок на сайте работать перестанет
- некоторое снижение производительности (лишний запрос)

Вместо этого нужно использовать включаемые области.

**Отправка отзывов о товаре** не выделена в отдельный компонент. Это повышает нагрузку, снижает возможности поддержки и развития.

**Редактирование товаров “С лица”** не работает . Штатная и очень удобная возможность системы 1С-Битрикс, появившаяся несколько лет назад, не подключена. Редактирование товаров для контент-редактора необоснованно усложнено.

#### **По стилю кодирования:**

Перебегающие блоки php, html, js кода. Код разного типа не вынесен в отдельные файлы, недостаточно применяется шаблонизация.

Поддержка такого кода, поиск ошибок и добавление функций сильно затруднены. Смена дизайна будет практически невозможна без переписывания кода.

#### **В целом:**

Не оптимально, неряшливо, структурно неправильно.

Такой стиль невозможно списать на то, что сайт в процессе разработки.

Причиной является, вероятнее всего, отсутствие опыта работы в команде, где приняты определенные стандарты кодирования и правила работы с 1С-Битрикс, рекомендованные разработчиками. Разработчик, вероятно, просто не знает как правильно, или крайне торопится.

#### **Вообще по культуре программирования:**

много “мертвого”, неиспользуемого, оставленного “на всякий случай” кода. В некоторых местах используется запрещенный к использованию морально устаревший (примерно 7 лет назад замененный на другой стиль) синтаксис, например `$val[GUID_VALUE]` и подобные устаревшие конструкции.

Много файлов “\*\_old.php”, оставленных “про запас”.

Неправильно используются специальные константы, например некоторые определяются после подключения пролога.

В частности, в файле `ajax/addToCart.php` эффекта от объявления константы нет. Пример:  
`require($_SERVER["DOCUMENT_ROOT"] . "/bitrix/modules/main/include/prolog_before.php");`

```
define("NO_KEEP_STATISTIC", true);
define("NOT_CHECK_PERMISSIONS", true);
```

Входные данные часто не проверяются, обработки некорректных состояний данных нет.

Пример. Код написан так, что может вызвать как минимум появление пачки сообщений об ошибках, как максимум -- неправильную работу из-за отсутствия или неправильного применения проверок данных (места нарушений выделены)

```
if($_REQUEST['ELEMENT_ID'] && !empty($_REQUEST['ELEMENT_ID']) && empty($_REQUEST['PRODUCT_XML_ID'])){

    $arResult = Array("ID", "XML_ID",);
    $arResult = Array("IBLOCK_ID" => 30, "ID" => $_REQUEST['ELEMENT_ID']);
    $res = CIBlockElement::GetList(
        Array(),
        $arResult,
        false,
        false, $arResult);
    while ($obj = $res->GetNextElement()) {
        $arResult = $obj->GetFields();
    }
    $arResult=$arResult['XML_ID'];
}
```

В каталоге /ajax/ находятся файлы-обработчики ajax-запросов. Логика обработки запросов реализована непосредственно в них. Хорошим стилем считается выносить логику в компоненты и модули.

Последствием этого является:

- потенциальная уязвимость безопасности, когда прямым обращением к обработчику может быть получен несанкционированный доступ
- затруднение возможностей поддержки кода
- результат не кешируется
- при наличии более чем одного ajax-обращения на странице все будет сильно тормозить

## Производительность

На страницах есть некешированные компоненты. Кастомизированные компоненты не кешируются (пример: `itcoins:product.properties`).

На эталонной виртуальной машине битрикса (производительность: 30) выполнение `/ajax/getLinkedEl.php` занимает в среднем 0.7, максимум 2.5 сек. Этот файл вызывается при показе каждой карточки товара,

после ее загрузки (в среднем 0.6 сек.). Для комплектов вызывается дважды, последовательно. Т.о. детальная страница товара открывается почти секунду или больше. Количество товаров и связанных с ними данных вырастет, приводя к увеличению этого времени.

Нормальным является генерация любых страниц сайта за время 0.1-0.3 секунды. 0.5 секунды для сложной страницы является предельно терпимым показателем.

Время 0.7-2.5 секунды получилось без всякой нагрузки на сайт, в отсутствие импорта из 1С и посещаемости.

Вывод: быстроедействие неудовлетворительное.

## Интеграция с 1С

На текущий момент на сайте настроена типовая загрузка товаров из 1С и ручная загрузка дополнительных данных через CSV по инструкции.

Учитывая что отсутствуют какая-либо автоматизация, какие-либо индивидуальные технические решения, анализировать нечего.

Тем не менее, текущая реализация интеграции с 1С видится неоптимальной из-за:

1. ручного переноса дополнительных данных через CSV. Импорт этих данных технологически возможен путем доработки модулей обмена в 1С и на сайте.
2. схемы данных, при которой свойства распределены по отдельным инфоблокам. чтобы построить карточку товара, нужно сделать множество достаточно ресурсоемких запросов. сведение всех свойств в один инфоблок позволит сократить число запросов до единиц

## Выводы

Качество программной реализации низкое. Нет культуры и опыта разработки под 1С-Битрикс, стандарты не соблюдаются.

Оставлены уязвимости, места повышенной нагрузки. Поддержка и развитие этого кода невозможны.

Из предложенного на аудит программного кода ничего нельзя использовать повторно.

Что касается структуры данных, полученных из 1С, и в целом реализации этого механизма. Применен стандартный импорт, дополнительные данные грузятся вручную. Этот механизм может быть использован и далее с переделкой распределения свойств по инфоблокам (сведения многих свойств в один инфоблок) для повышения производительности.

24 января 2014 года

\_\_\_\_\_ Ведущий разработчик отдела сопровождения сайтов,  
сертифицированный специалист по Bitrix Framework  
Павел Машанов

\_\_\_\_\_ Руководитель отдела управления проектами,  
Алексей Шкарупа

\_\_\_\_\_ Ген. директор, к.т.н.,  
Степан Овчинников