# Отчёт о тестировании сайта shop.wildorchid.ru

# **1.** Первичный аудит сайта

## 1.1. «Самописные» компоненты

В основном, компоненты модифицированы из-за внедрения нестандартного функционала, что логично.

Некоторые компоненты можно привести к штатному виду, а дополнительную логику вынести в специальные файлы. Но к каждому «кастомизированному» компоненту нужно подходить отдельно, потому что любая строчка кастомизации тянет за собой целый ворох зависимостей.

Вот часть «кастомизированных» компонентов:

* Баннеры.
* Детальное отображение элемента каталога.
* Сортировка каталога.
* Список товаров каталога.
* Умный фильтр каталога.
* Регистрация пользователя.
* Профайл пользователя.
* Все виды корзин пользователя.
* Страница поиска.
* Выбор локации.
* Компонент восстановления пароля.
* и прочие.

## 1.2. «Самописное» ядро

По адресу shop.wildorchid.ru/bitrix/php\_interface/breadhead/ есть большое скопление каталогов и файлов, вкупе они представляют из себя ядро, еще одно ядро проекта с кучей «самописных» функций.

## 1.3. Функции практически не документированы.

В ходе разработки функционала покупки в один клик выяснилось, что на сайте переписаны не только системные компоненты, но и некоторые системные функции. К примеру магазин не поддерживает работу с штатным функционалом почтовых событий, вместо этого написан целый класс, обрабатывающий почтовые события магазина, переписано все вплоть до обращений к БД. Сложно сказать зачем это было сделано, но штатными методами магазин почту больше не отправляет.

Помимо почты переписаны функции обработки заказов, работы с корзиной, купонами, методами оплаты и доставки, регистрации и авторизации пользователей.

В отдельное ядро вынесена дополнительная бизнес-логика, возможно часть функционала можно было реализовать на штатных механизмах системы, но сейчас перевести все на штатные рельсы будет крайне сложно, тысячи строк кода не дадутся без боя.

## 1.4. Ситуация с ID и номерами заказов

Ситуация скверная, в системе поддерживается хрупкое искусственное равенство между ID и номером заказа. При этом повсеместно вперемешку используются и ID и номер заказа. К примеру, в почтовых уведомлениях пользователю отправляются ID заказа, а в ЛК выводятся номера, в 1С отправляются номера заказов, эквайринг опять работает с ID заказа, а не с его номером. Из-за этого любое расхождение номера заказа с его ID влечет за собой множество бед - неразбериха у пользователей, менеджеров и бухгалтерии.

Советую избавиться от ID, оставить его на откуп 1С-Битрикс, а пользователям, менеджерам и 1С работать с номером заказа. В крайнем случае программно зафиксировать номер заказа за его ID, в таком случае рассинхронизации не будет, т.к. номер заказа будет построен на базе его ID.

## 1.5. Папка - /local/

По стандартам 1С-Битрикс, весь «кастом» должен выноситься из папке /bitrix/ и находиться в /local/. Сейчас в этой папке лежит часть выгрузки в Яндекс.Маркет, ей там явно не место.

Нужно перенести все компоненты/кастомное ядро/модули и шаблоны в /local/ и забыть про /bitrix/ в этой папке вести разработку запрещено.

Также в корне сайта нашлась папка /layout/ содержащая исходники макетов, исходники не закрыты от индекса. Часть содержимого папки нужно перенести в /local/, другую часть либо скрыть от индекса, либо вообще удалить с реального сайта.

## 1.6. Страницы и разделы в корне сайта которые нужно скрыть либо удалить

Некоторые страницы представляют из себя тестовые выборки или какие-то сервисные инструменты.

Также есть потенциально опасные скрипты, к примеру скрипты рефакторинга в разделе /services/.

Помимо этого, есть и интересные файлы, к примеру выгрузка пользователей со всеми данными.

* shop.wildorchid.ru/bx\_1c\_import.php
* shop.wildorchid.ru/111.php
* shop.wildorchid.ru/im\_test.php
* shop.wildorchid.ru/geo.php
* shop.wildorchid.ru/im\_test.php
* shop.wildorchid.ru/test.php
* shop.wildorchid.ru/test2.php
* shop.wildorchid.ru/requests/
* shop.wildorchid.ru/rsb\_acquiring/...
* shop.wildorchid.ru/service/...
* shop.wildorchid.ru/survey/…

Просто закрыть или удалить большинство файлов и директорий нельзя, т. к. часть из них точно используется кастомным ядром, часть используется при обмене с 1с. Нужно тщательно изучать каждый из файлов и принимать решение в индивидуальном порядке.

## 1.7. Обновление платформы

При обновлении платформы с 15.хх на 16.хх встанет вопрос о совместимости кастомного ядра, того что сейчас расположено в shop.wildorchid.ru/bitrix/php\_interface/breadhead/. Переход на 16.хх версию повлечет за собой полное обновление движка магазина с конвертацией таблиц БД и данных, часть методов API будут помечены как устаревшие и в последствии перестанут поддерживаться платформой.

Кастомное ядро местами напрямую обращается к БД сайта, а значит велика вероятность поломки после обновления ядра и БД. Здесь можно посоветовать по частям выпиливать куски кастомного ядра которые напрямую обращаются к БД и переписывать их при помощи нового API и ORM (ядра D7).

## Общее заключение

Система уже на данный момент является трудноподдерживаемым монстром, внесение даже малейших правок влечет за собой большие трудности с отловом всевозможных ошибок и неявного поведения системы из-за «кастомного» ядра.

Систему нельзя обновлять. 24.03.2016 в релиз вышло ядро 16й версии, а вместе с ним и новый интернет магазин, нам эти обновления не светят до тех пор, пока используется «кастомное» ядро.

На сайте есть много потенциально опасных файлов и директорий доступных для любых пользователей. С этим можно бороться, но чтобы ничего не сломать, придется потратить довольно большое количество времени на выявление всех зависимостей для каждого из файлов.

От части «самописных» компонентов можно избавиться, но к каждому нужно подходить отдельно и анализировать причину «кастомизации» и зависимость от «кастомного» ядра.

ID и номера заказов — хрупкой равновесие было восстановлено, но лучше бы обезопасить себя и использовать что-то одно.

## 2. Аудит работы системы кеширования

На сайте включен автоматический и управляемый кеш.

Кеш работает практически во всех разделах сайта, об исключениях далее.

В тоже время на каждую страницу приходится минимум по 45 запросов к БД, как селектов, так и апдейтов. В основном это обращения модуля статистики, он читает и пишет статистику пользователя. Также не обходится без запросов от кастомного ядра.

### 2.1. Периодически система кеширования «глохнет» и перестает функционировать.

Для тестирования проблемы полностью скопировано ядро текущего сайта на dev, пробовали подключать реальную бд, играться с конфигами сайта, итог нулевой, на деве кеш работает в штатном режиме.

После созвона с Виктором перезапустили апатч, вернули конфиги в исходное положение и кеш заработал.

Посмотрели конфиги основного сайта, пришли к выводу что система кеширвоания скорее всего просто забивает доступный стек памяти и после этого не функционирует. В тестовом режиме подкрутили настройки модуля кеширования, параметры увеличили в два раза. В течение недели система кеширования не «глохла».

### 2.2. Разделы и блоки исключения, в них кеш по тем или иным причинам отключен

Весь пользовательский раздел, корзина и список желаний, тут информация не кешируется чтобы быть всегда актуальной, с этим можно бороться, н опока бессмысленно.

Самописное ядро shop.wildorchid.ru/bitrix/php\_interface/breadhead/ не кеширует множество запросов к БД, во многом это связано с тем что идут запросы которые нужно обработать здесь и сейчас, это статистика, заказы, регистрация и прочие запросы которые нельзя кеширвоать, но помимо них встречаются и те что можно было бы обернуть в кеширующие функции обертки.

Баннеры - на всех страницах в верхней части подключается три компонента для вывода баннеров. Два штатных компонента и самописный из неймспейса breadhead. Во всех трех компонентах есть некешируемые запросы. Походив по страницам сайта я не нашел ни одной где бы хотя бы один из компонентов выводил баннеры, возможно компоеннты стоит отключить и сэкономить десяток запросов в БД.

**Самый проблемный некешируемый раздел:** http://shop.wildorchid.ru/specials/

В ходе изучения проблемы, оказалось, что информация на страницу выводится следующим образом - основной компонент вывода - комплексный компонент новостей, каждая из страниц это элемент инфоблока "тематические страницы", в каждой детальной карточке элемента ИБ выводится «самописный» компонент секции каталога. Компонент новостей кеширует детальную карточку элемента ИБ, он ведь не знает что внутри деталки есть еще один компонент. «Самописный» компонент вывода списка элементов секция в свою очередь выводит товары, но только для первого хита, потом он не отрабатывает из-за того что данные полностью закешированы. AJAX\_MODE также не поддерживается.

По хорошему, нужно избавиться от прослойки в виде элементов ИБ, параметры выборок перенести в секции и выводить товары именно при помощи секций, это позволит избавиться от одного промежуточного звена и проблем с кешем.

### 2.3. Кеширвоание картинок товаров

protected static function func\_imagemagick\_img\_resize(){}

shop.wildorchid.ru/bitrix/php\_interface/breadhead/service/tools.php строка 270

Это функция которая выполняет ресай изображений и сохраняет результат своей работы в папку.

Вроде все нормально, ресайзы делаются, но папка с каждым днем увеличивается в размерах.

Проблема заключается в том что про эти ресайзы ничего не знает система (ядро) и система кеширования тоже. Ресайзы попавшие в папку никогда от туда не удаляются, потому что нет системы учета файлов, система не знает нужен ли ей конкретный файл или его следует удалить потому что товара или изображения для которого был сделан ресайз уже давно нет.

Функцию можно модернизировать, сохранение и хранение файлов перенести на штатные рельсы системы, тогда файлы будут удаляться при сбросе кеша и обновляться при необходимости.

## Общее заключение

**С системой кеширования есть проблемы, как с самой системой в целом, так и в ее использовании в разных частях сайта.**

**Каждый из проблемных участков рассмотрен в аудите, практически каждый из них является отдельной большой задачей.**