

РАЗРАБОТКА ПОД E-COMMERCE\*, MAGENTO\*

## Magento 2. Le Roi est mort! Vive le Roi?

flancer 9 марта 2017 в 12:45  4,4k

Впервые с Magento (тогда еще "единичкой") я столкнулся лет, эдак, 6 назад. С тех пор так с ней и работаю, с большей или меньшей плотностью. Изначально пост хотел назвать "Quo vadis, Magento?", но, [как оказалось](#), этим популярным вопросом сообщество задавалось уже не раз — и когда Magento приобреталась ebay'ем, и когда продавалась, и все то время, пока делалась "двоечка", да и поныне этот вопрос остается актуальным (раз уж даже у меня возникло желание использовать такое название). Поэтому пост называется так, как он называется.



Под катом же я попытался сформулировать свое собственное (сиречь — субъективное) видение перспектив этой платформы — полное брюзжания и уныния. Без подробных выкладок. Без детальных размышлений. Без доказательств. И главное —

да, главное — без надежды.

## Le Roi

В моей практике Magento, пожалуй, является самым сложным web-приложением, с которым я сталкивался (еще раз акцентирую

внимание на **субъективности** своих выкладок!). Коллега [Jonathan Edwards](#) как-то выразился "programming is a desperate losing battle against the unconquerable complexity of code and the treachery of requirements" (*Программирование — безнадежно проигранная борьба с непреодолимой сложностью кода и вероломством требований*). Так вот, Magento в этой безнадежной борьбе продвинулось дальше всех остальных виденных мной web-приложений.

В основу Magento изначально были заложены принципы приспособляемости и расширяемости: EAV, modules, rewrites, events — вот те механизмы, которые позволили привлечь к Magento множество пользователей и, главное, разработчиков и занять главенствующее место в e-commerce. Основными "фишками" Magento были и остаются не только богатство функционала "из коробки", но и внушительное количество модулей сторонних разработчиков (платных и бесплатных), позволяющих донастроить магазин "под себя".

Magento стало центром притяжения с одной стороны множества девелоперов — можно было либо получить [хорошую рекламу](#), сделав небольшой, но нужный всем бесплатный модуль, либо получить [хорошие деньги](#), сделав коммерческий модуль под определенные потребности клиентов, не охваченные базовым функционалом (или охваченные только Enterprise-версией). Сейчас на Magento Connect выставлено не менее [966 страниц](#) модулей для Magento 1, по 10 штук на страницу — это почти 10,000 расширений.

С другой стороны Magento стало центром притяжения и для пользователей — они сразу же получали не только работоспособный магазин, но могли также подобрать для него привлекательную тему, расширить функционал с помощью готовых модулей. Или заказать создание "своих" модулей. Или "своей" темы. Расширяемость была одной из притягательных маркетинговых фишек Magento. Конечно, в реальности добавление/изменение функционала было не совсем **тривиальной задачей**, но тем не менее, такая возможность была изначально добавлена в платформу при ее создании и успешно эксплуатировалась при оценочном сравнении с конкурентами. В результате количество магазинов на Magento достигло по некоторым оценкам 2015 года почти четверти миллиона.





Это двухстороннее притяжение создало своеобразную экосистему, в которой создавались, опробывались, приживались или

выкидывались различные решения в области e-commerce на базе платформы Magento. В первой версии не было полноценного тестирования, но его отсутствие компенсировалось количеством установок — если использовать общеупотребимый функционал и не выстраивать экзотических конфигураций, то вероятность столкнуться с ошибками в работе магазина была довольно низкой. Определенное количество существующих **крупных магазинов** также свидетельствовало о неплохих возможностях платформы — по крайней мере, с точки зрения среднего и мелкого бизнеса.

В целом, еще совсем недавно Magento являлась безоговорочным лидером на рынке e-commerce.

### ... est mort

Так почему же лично я смотрю на будущее Magento с пессимизмом? Наверное по той же самой причине, по которой с пессимизмом смотрели на будущее в машинном отделении "Титаника". Возможно, даже капитан Эдвард Джон Смит, к которому стекалась вся информация, не понимал поначалу всей трагичности ситуации, не говоря уже о пассажирах, которых аналоги маркетинговых служб того времени убедили в непотопляемости "Титаника". Но в машинном видели и лопнувшие заклепки, и щели между стальными листами, в которые, klokочa и пенясь, врывалась ледяная морская вода, и безуспешную работу насосов. В машинном точно не знали, сколько еще "Титаник" сможет продержаться на поверхности, но то, что до Америки он не дотянет — знали наверняка.

Magento 2, без сомнения, является развитием предыдущей версии — улучшилась модульность, достигнута совместимость с composer'ом, используются более современные решения как на "фронте", так и на серверной стороне, код [общедоступен](#) и [тестируем](#), производительность улучшена. На первый взгляд вполне себе замечательные перспективы.

Но что же, все-таки, омрачает картину?

## Объем

В Magento 2 — 1,357,121 строк PHP кода и 2,449,066 строк всего (WordPress — [423,759](#); WooCommerce — [131,549](#); osCommerce — [208,928](#); Drupal — [704,269](#)).

### [Magento 1.9.3.2](#)

### [Magento 2.1.5](#)

В самом количестве строк нет ничего ужасного (тот же SugarCRM содержит [2,149,457](#) строк) — это просто показатель "инерционности" приложения, аналог массы в физике, причина, по которой корабль не может резко сменить курс. В чем же именно проблемы?

## Структуры данных

У разработчиков Magento 2 был замечательный шанс значительно переработать структуру данных — было официально объявлено,

что вторая версия будет несовместима с первой. И по структурам данных сделана значительная подвижка — переименованы/добавлены/удалены таблицы и поля так, что база данных от первой версии действительно не могла быть использована во второй.

Но, видно что основным мотиватором при переработке все-таки был "как можно меньше изменений". Иначе чем объяснить мигрировавшую в "двойку" нестыковку соответствия `store_id` в БД и Store в админке (`store_id` в БД атрибут Store View в админке, а вот `group_id` в БД — как раз и является атрибутом для Store в админке — и это нельзя понять, это нужно запомнить)? Или неоднозначные зависимости между данными в тех же store-таблицах или [stock-таблицах](#)?

Или такой фундаментальный (или риторический?) вопрос — почему для EAV заложено только 8 типов сущностей (`customer`, `customer_address`, `catalog_category`, `catalog_product`, `order`, `invoice`, `creditmemo`, `shipment`) из которых через админку для расширения доступны только атрибуты продукта (`catalog_product`)? Почему проигнорированы такие сущности, как `admin`, `authorization`, `stock`, `sales_rule`, `cms`, `rating`, `report`, `review`, ...?

Это, кстати, здорово сбивает, особенно поначалу — зачем плодить в структурах данных поддержку для EAV, если по факту конечный пользователь Community-версии может его применить только к одному типу данных, к продуктам? И да, это тормозит все приложение (как минимум в developer-режиме) и является причиной появления flat-таблиц. И да, в "двойке" часть атрибутов



клиента все-таки перенесли в `customer_entity` (28 полей, против 12 "единички").

## Код

В Magento 2 значительно переработали архитектуру — и это явный прогресс. Я могу только поприветствовать появление DI и слоя классов-репозитория, интеграцию API со [Swagger'ом](#) и расширение CLI-набора команд собственными командами. Но... попробуйте программно сохранить store (Store View в терминах админки), руководствуясь [best practices](#) от Magento 2 — [не получится](#). Это при том, что репо-класс для `\Magento\Store\Model\Store` [существует](#), правда без метода `save()`, а стандартный для "единички" метод `save()`, наследуемый от `\Magento\Framework\Model\AbstractModel`, [объявлен @deprecated](#).

А попробуйте найти имплементацию интерфейса [\Magento\Sales\Api\OrderStatusHistoryRepositoryInterface](#). Как бы должен быть `"\Magento\Sales\Api\Data\OrderStatusHistory\Repository"`, согласно [di.xml](#). Хорошо, попробуйте найти класс `"\Magento\Sales\Api\Data\OrderStatusHistory\Repository"`. Может быть этот интерфейс нигде не используется? Нет, [используется](#). Может автогенерится? Нет, [генерятся](#) Factory, Proxy и Interceptor'ы, а у нас — Репозиторий.

Побродите по коду — есть весьма переработанные модули (customer, например), а есть обернутый код от "единички". Я понимаю, что "двойка" делалась на базе "единички", что было

принято "политическое" решение использовать как можно больше существующего кода, чтобы выпуститься к какой-нибудь знаменательной дате (например, продажа ebay'ю или ebay'ем, привлечение очередных инвестиций и т.п.). Или чтобы "сократить затраты на разработку". Беда не в этом. Если бы только это, то ситуацию можно было бы выправить путем планомерной и методичной переработки кода. Беда в том, что сейчас в Magento рулят "эффективные"/"кризисные" менеджеры, а не технари (ребята, без обид — я не знаю ваших имен, я сужу только по результатам).

## Релизы

Ну а что я еще могу думать, когда через 2 недели после 2.1.4 выходит 2.1.5 единственное [назначение](#) которого:

*This release updates the copyright date in every file. It does not contain any functional changes or security improvements. Isolating these changes in a single release is intended to simplify future updates and developer workflow.*

Релиз! Целый релиз несущественных с точки зрения технаря изменений!!! И что теперь, объяснять владельцам магазинов, что работающая у них 2.1.4 с функциональной точки зрения точно такая же, как 2.1.5, разве что только не последняя?

А [план релизов](#) на середину февраля:

For now 2.1.5 will be copyright, 2.1.6 security and 2.1.7 will have bugfixing.

Может еще bugfix'ы разобьем на две группы — для фронта и для серверной стороны? И под каждую свой релиз сделаем? А если вдруг на следующий день после выпуска 2.1.6 будет обнаружена дыра в security, то мы ее "залечим" в 2.1.8 или в 2.1.9?

## bugfixes

Причем самое печальное, IMHO, это то, что в версию 2.1.5 не вошел вот [этот](#) pull request. Вот ошибочный код:

```
public function getStockId()
{
    return $this->getData(self::KEY_WEBSITE_ID);
}
```

вот исправленный:

```
public function getStockId()
{
    return $this->getData(self::KEY_STOCK_ID);
}
```

Его запостил коллега [Félix Delval](#) еще 29 сентября 2016 года. До выхода версии 2.1.2 (12 октября 2016 года), версии 2.1.3 (14 декабря 2016 года), версии 2.1.4 (7 февраля 2017 года) и до пресловутой версии 2.1.5 (21 февраля 2017 года). Судя по плану релизов, ошибку исправят в 2.1.7 — месяца через 3-4.

А ведь это явная ошибка, понятная даже начинающим программистам. Сколько она времени сожрала у всех 10-ти (!!!) участников обсуждения этого бага и до сих пор не исправлена!

Ой, нет, вру — [исправлена](#) в developer-ветке. А вот в [релизах](#) — так и продолжает оставаться.

Я не задаю вопрос "сколько еще таких *исправленных* багов есть в developer-ветке и нет в релизных", просто потому, что боюсь услышать ответ.

## Инерция

Magento имеет хорошую историю, внушительное сообщество, приличный "кусочек" в пироге e-commerce, но... вместе с тем я наблюдаю потерю управляемости в проекте. Magento 2 перестала своевременно реагировать на изменения. Она стала слишком большая. Как Microsoft в свое время.

Возможно, вступают в действие какие-то законы, по которым не только программирование, а и вообще создание сложных систем — "a desperate losing battle". Структура вплотную подходит к какой-то неизвестной пока еще границе сложности, когда каждый следующий шаг стоит как все предыдущие вместе взятые, и для наращивания функциональности проще создать нечто новое — другую структуру, которая реализует аналогичный функционал по другим принципам и на меньшем уровне сложности и позволит этот функционал нарастить, пока сама в свою очередь не приблизится к той же самой границе. В физике есть подобная

граница — скорость света, почему бы ей не быть и в программировании?

Так вот, инерция в Magento 2 в силу ее сложности стала настолько большой, что управлять ее разработкой становится все труднее и труднее. Вышеописанные проблемы в структурах данных, коде, bugfix'ах, релизах — это следствия приближения к "верхней границе сложности".

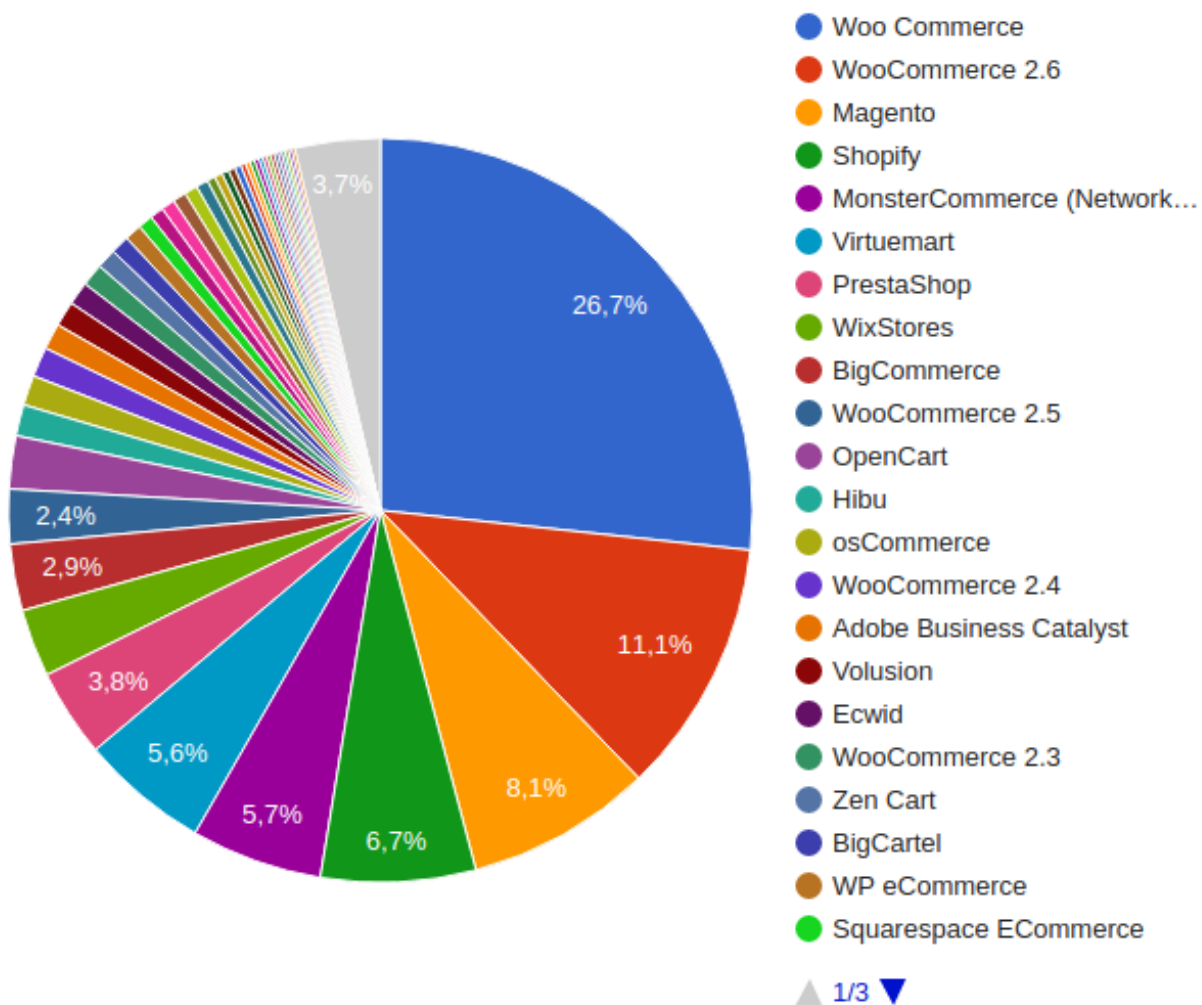
## Disclaimer

Эй, это мое субъективное мнение. Я не юрист и не эффективный менеджер, я — разработчик, технарь. Я вижу картину со своего ракурса и вижу ее вот так.

## Vive le Roi?

Как говорится, "свято место пусто не бывает". И если не Magento, то — кто?





Судя по [этим](#) данным — WooCommerce? Или пока еще не выстреливший [OroCommerce](#) — от авторов "единички"? Или может вообще малоизвестный игрок, вроде [Sylus](#)? Или новый "король" вообще будет не на [PHP](#)? Куда перетекут конечные пользователи e-commerce платформ и кто станет новым центром кристаллизации?

На все эти вопросы у меня ответа нет.

Проголосовать:



+7



Поделиться:



Сохранить:



Комментарии (14)

## Похожие публикации

**Крупнейшая битва в истории EVE Online:  
уничтожено кораблей на \$200 000**

alizar • 29 января 2014 в 12:16

329

**С борта МКС впервые по лазерному каналу была  
передана широкополосная информация на  
наземный пункт**

imwode • 4 октября 2012 в 15:54

63

**Galaxy Tab был представлен широкой публике на  
IFA**

lamo4ok • 2 сентября 2010 в 14:50

153

## Популярное за сутки

## Яндекс открывает Алису для всех разработчиков. Платформа Яндекс.Диалоги (бета)

BarakAdama • вчера в 10:52

69

## Почему следует игнорировать истории основателей успешных стартапов

ПЕРЕВОД

m1rko • вчера в 10:44

20

## Как получить телефон (почти) любой красоты в Москве, или интересная особенность MT\_FREE

ИЗ ПЕСОЧНИЦЫ

sab404 • вчера в 20:27

24

## Java и Project Reactor

zealot\_and\_frenzy • вчера в 10:56

10

## Пользовательские агрегатные и оконные функции в PostgreSQL и Oracle

erogov • вчера в 12:46

6

## Лучшее на Geektimes

## Как фермеры Дикого Запада организовали телефонную сеть на колючей проволоке

NAGru • вчера в 10:10

31

## Энтузиаст сделал новую материнскую плату для ThinkPad X200s

49

alizar • вчера в 15:32

## Кто-то посылает секс-игрушки с Amazon незнакомцам. Amazon не знает, как их остановить

85

Pochtoycom • вчера в 13:06

## Илон Маск продолжает убеждать в необходимости создания колонии людей на Марсе

139

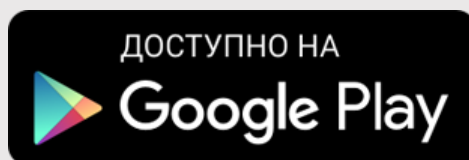
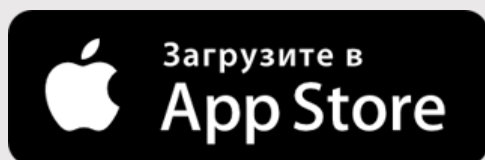
marks • вчера в 14:19

## Дела шпионские (часть 1)

16

TashaFridrih • вчера в 13:16

Мобильное приложение



Полная версия

2006 – 2018 © TM