*Маркетологи проанализировали собранные данные и определили несколько аналитических моделей, которые, по их мнению, могут поднять продажи или улучшить восприятие клиентами процесса выбора и покупки. С этими моделями маркетологи обратились к владельцам бизнеса. Те в свою очередь боятся, что станет только хуже. Вас просят предложить решение для проверки гипотезы маркетологов на тестовой группе клиентов.*

Поднять продажи можно либо менеджерско-административными (обучить людей и технику впаривать товары активнее), либо услуго-скидочно, тоесть добавлять «вес» покупке определенным товарам доп.услугами, будь то доставка на вертолетах или упаковка в розовый картон. Или как-то еще, но эти механизмы мне неизвестны, нужны профильные спецы.

Если второй вариант – только добавить уже заранее выданный бизнесом визуал про розовый картон на сайт.

При первом – возможно усиление логики и включение сложных моделей поведения в тот же сайт (иные аспекты системы с клиентами не связаны, контакт только через сайт.) Иными словами – необходимо разбивать модуль «сайт» на более сложную конструкцию с обработкой логики. Возможно, расширение моделей и БД для сервисов Клиент, Продукт, Заказ, чтобы лучше соответствовать маркетину, но тут всё слишком зависит от «хотелок». Можно и управление дронами прикрутить, если будут деньги и иные ресурсы.

Предположим, что маркетинг «изобрел» демонстрацию товара на основе старых заказов, сложный расчет предподчтений на основе ранее «просмотренных» товаров и совершенно случайно добыл доступ к БД Гугла и «ЗНАЕТ», что именно пользователь ищет еще.

1. *Предложить стратегию такого решения. Описать возможные варианты того, как это может быть реализовано технически, обосновать выбор.*

Было так:



И бекенд занимался лишь тем, что проверял входящие-исходящие данные, ретранслировал запросы Фронта и отдавал ему полученные данные из иных сервисов. Так же – запоминал действия пользователя и отдавал их сервисам логирования.

Как предлагаю с учетом вышеописанных предположений:



Задачи бекенда не изменятся, серьезно его переписывать не придется, только добавить несколько интеграций.

Плюсы:

* Неограниченное число возможных стратегий для подключения, вопрос только в мощности
* Безопасность. Где сложная логика с данными, зависящими от пользователей, том вполне вероятны ошибки и прочие ошибки. Сломанный доп.обрабочик всё равно позволит бекенду отдать стандартную картинку.
* Асинхронность и быстродействие. Сложная логика может вычисляться долго. И негоже заставлять пользователя ждать. Не обязательно обрабатывать данные в режиме реального времени, ничего не произойдет, если он увидит рекомендуемые «именно ему и именно вот только 5 минут смс огромной скидкой только для вас» не прямо сейчас, а через несколько страниц, пока данные нормально обработаются. Запрос в ту же БД Гугла может выполняться ОЧЕНЬ долго, особенно если интернет тормозит и выборка приходит ОЧЕНЬ большая.
* Локальные БД позволят не гонять туда-сюда огромные массивы данных, а хранить на месте, просто переращивая при получении новых данных.
* Не особо поменяется логика фронта, все то же взаимодействие со своим беком, без сотен точек взаимодействия с кем бы то ни было еще.

Минусы:

* Каждый доп.сервис – точка отказа сам по себе, плюс интеграции как потенциальные проблемы.
* Сложная логика затратна по ресурсам (мощностям, времени  
   и т. д.)
* Сложная логика чувствительна к форматам данных, возможно придется подгонять иные одно к дугому.
* Возможно, данные будут обрабатываться долго за счет возможного последовательного обращения к нескольким доп.сервисам.

Т.к. должна быть тестовая группа – 2 варианта.

* Случайная выборка. При первом заходе на сайт, условно, с вероятностью 0.1 Клиент может быть помечен как обрабатываемый по-новому. Так же 10% от уже имеющихся Клиентов так же пометить. При наличии этой метки – использовать все сервисы… Или какие-либо конкретные и ставить несколько типов меток, смотря какую задачу поставит бизнес.
* Ручная выборка. Дать менеджерам инструмент, чтобы они руками включали эти механизмы у «проверенных» Клиентов.

Я за рандом. Исключит человеческий фактор при создании выборки. Повысит доверие к эксперименту.

Так же важно учесть. Есть 2 сайта: для клиентов и для менеджеров. Слишком разный функционал должен быть там реализован и модули Бэка и Фронта были бы огромными и неповоротливыми. Тем не менее, много кода дублируются.

Очень пригодится Chassis подход: шаблоны с нужным функционалом будут подключаться как модули, что позволит сохранить оперативность (не придется гонять данные туда-сюда по сети), повысить безопасность (сторонний сервис может умереть и повесить Бэк. В 2 раза больше шансов, что что-то пойдет не так), не дублировать код. Глупо писать проверку входящих данных для нового Заказа дважды – а так подключил шаблон, весь функционал «из коробки». Или коды ошибок и реакция на них (получил ошибку, что сервис «мертвый» - спроси еще раз.

1. *Указать, какие сервисы будут затронуты в результате такого подхода и почему.*

В любом случае будут серьезно дописан Бекенд сайта. Также Фронт сайта. Новые механизмы нужно внедрить, позволить собрать данные и т.д.

Из иных сервисов – возможно добавятся поля в моделях Клиента, Заказа, Продукта. Тут всё зависит от поставленной задачи и собираемых данных. Возможно, придется добавить некоторые методы специально для этих сервисов. Например, «Дай данные по всем Заказам клиента за период времени с включением всех Продуктов».

Логирования нужно дорабатывать, чтобы собирались метрики по новым сервисам, новому поведению пользователей, велась маркировка клиентов, обрабатываемых «старым» способом и «новым».

1. *Описать, изменится ли формат данных для анализа новых данных. И если да, то как.*

Всё зависит от того, какие именно метрики будут нужны. Если добавятся какие-то специфифичсескиек форматы (если пользователь задержал мышь на товаре дольше, чем 0.3 секунды – добавить товар в рекомендуемые на 1 место), что раньше не делалось – придется это начинать хранить и обрабатывать.

В теории, если новые стратегии основывались на определенных данных, то, скорее всего, эти же данные и должны использоваться в логике работы, потому форматы серьезно меняться не должны.

*После того, как вы представили своё видение решения поставленной задачи, к вам обратились администраторы. Они говорят, что решение усложнило их работу: отслеживать конфигурацию инфраструктуры стало труднее, нужно больше людей, чтобы управлять этим расширяющимся «зоопарком» сервисов.*

*Заливать проблему деньгами — не выход, вы понимаете, что бюджет ограничен. Подумайте, что можно предложить администраторам в качестве решения.*

1. *Укажите, управление какими сервисами можно автоматизировать, а какие оставить как есть, с ручной поддержкой. В каком порядке вы предложите реализовывать такой переход, и почему? Какие риски вы видите в автоматизации процессов?*

Автоматизировать можно и нужно сервисы, отвечающие за хранение и выдачу данных, такие как «Клиент», «Заказ», «Продукт». Можно на них повесить автоматическое развертывание реплик, проверку их работоспособности, отключение «мертвых» и т.д. Всегда будет как минимум по 2 экземпляра каждого сервиса, что позволит избежать задержек и каскадного умирания системы из-за невозможности получить данные. Сервисы логирования в ту же копилку. Всегда должно жить хотя бы 2. Если минуту занято больше 80% мощности у всех сервисов – развернуть еще 1 экземпляр, через 1 минуту после развертывания повторить и так пока не упадет нагрузка до 50% либо копий не станет 5. Это магазин мебели, даже 3 сервиса загружены будут «никогда», просто я немного паникер и перестраховщик. Если и 5 загружены – скорее всего это зерг-раш или кто-то систему ддосит. Если нагрузка меньше 5% у всех в течение 10 минут – убить первый последний развернутый.

Затраты на эти сервисы не сильно большие, ресурсов много не нужно. Наверняка есть бесплатные решения с приемлемым функционалом.

Новые сервисы, имеющие сложную логику, масштабить сложно, ибо они весьма требовательны по ресурсам, это может быть просто дорого (если логика действительно сложная, предполагаем, что прям нейросети). Потому только мониторинг, возможность подхватывать результаты из БД и оперативное развертывание в случае смерти (с возможностью чистить последние пришедшие данные и записи в БД, вдруг именно эти данные систему и крашнули). Теоретически, даже потери данных не критичны: пользователь не поймет, что что-то случилось, ведь он увидит «стандартную» картинку и весь функционал будет работать.

По порядку – сначала самый частосправшиваемый «Продукт». Его спрашивают часто, много, все подряд. Оперативная реакция на изменения высвободит часть ресурсов. Затем – Заказы и Клиенты. Это повысит общую отказоустойчивость и производительность системы, уже высвободит часть ресурсов.

Сервисы логов – по настроению и если текущие не справляются. Даже если он умрет – просто будет копиться Очередь и данные актуализируются после переподнятия сервиса.

Сервис Очереди важен, возможно, следует предусмотреть механизмы реакции на переполнение, и иные проблемы, но это нужно смотреть в настройки конкретного продукта. Наверняка все в заводских решениях не раз и не два продумано. В любом случае, автоматизировать можно лишь процесс поднятия и уничтожения, решение – только за человеком. Единственное, при потере очереди (например, при развертывании нового экземпляра) должен быть механизм автоматического согласования данных. Чтобы актуализировать статусы заказа (мало ли, уже доставили или отменили.)