



РОБОТЫ РАБОТАЮТ

Игорь Антонов

Максим Зарудняк

РОБОТЫ РАБОТАЮТ

*История развития туроператора
«Библио-Глобус»
глазами сотрудников отдела
информационных технологий*

Оглавление

Оглавление	2
Вводное слово	3
Глава 1. Историческая часть	4
Запуск	4
Достижение лидерства	6
Кризис 2014	12
Восстановление	18
Глава 2. Прикладная часть	20
Технология	20
Автоматизация	25
Три шага	29
Экспансия	31
Безопасность	35
Глава 3. Роботы работают	37
Правила	37
ИТ-отдел	42
Сотрудники	49
Глава 4. Метаморфоз	52
Технический долг	52
Изменение модели	54
Рынок	68
ЗАКЛЮЧИТЕЛЬНОЕ СЛОВО	70

Вводное слово

Эта книга про автоматизацию большой компании и людей, способных делать сложные вещи в сложных условиях.

Эта книга для тех, кто хочет сравнить свой путь с нашим.

Эта книга — не учебник, это — история.

Лучше написать её самим, чем читать о себе в новостях.

Всё, что здесь описано, имеет только практическое значение.

Наш опыт учит уважать любое технологическое решение, если оно работает.

Мы любим роботов, они — работают.

Глава 1. Историческая часть

Запуск

Эта история началась в 2004 году с приходом в туроператорскую компанию «Библио- Глобус» новой команды разработчиков во главе с Игорем Антоновым. К тому времени владельцы компании — Александр и Юлия Туголуковы — пришли к печальному выводу, что имеющиеся программное обеспечение тормозит развитие компании, не позволяя масштабировать бизнес. Перед нашей командой ставилась задача вывести «Библио- Глобус» в лидеры туристического рынка, а для этого было необходимо построить всю инфраструктуру и программное обеспечение с нуля. У команды уже была наработана платформа для быстрого разворачивания проектов и план дальнейших действий.

Однако руководство компании, минимизируя риски, пошло на неожиданный шаг, создав для нас атмосферу жесткой конкуренции с предыдущей командой разработчиков, которая продолжала работать параллельно с нами. При таких условиях мы могли получить разрешение на полноценное внедрение лишь доказав преимущество нашей системы и ее перспективность.

В те годы туристическая отрасль только осваивала автоматизацию бизнес-процессов и не выделялась какими-то экстраординарными программными продуктами, масштабами систем бронирования или мгновенной реакцией на изменение ситуации на рынке. Небольшой туроператор вполне удовлетворялся Excel-таблицами, агрегаторы услуг только выходили на большой рынок, а типовым продуктом были локальные офисные системы бронирования, например, туристическое программное обеспечение «Мастер-Тур» от компании Megatek.

Как известно, туроператор, помимо обеспечения контрактных обязательств, должен ответить на запрос туриста: «сколько мне будет стоить поездка?». Ответить на один запрос можно было с помощью калькулятора. Но когда речь шла в «горячий» сезон о предложении для агентств множества отелей, вариантов проживания и перелета, то расчёт всех этих комбинаций выполнялся часами. Туроператоры в то время могли обновлять у себя на сайте цены максимум два раза в день.

Перед нами стояла задача разработать самый «быстрый» на рынке прайс-лист — технологию, способную в режиме реального времени пересчитывать стоимость тура в зависимости от изменения вводных данных (цен на гостиничные номера, авиабилеты и т. д.).

Нам дали зеленый свет на дальнейшую разработку, когда команда предоставила решение, рассчитывающее прайс-лист с тысячами результатов за несколько секунд. Для этого пришлось полностью погрузиться в специфику ценообразования и, изучив его досконально, построить универсальное решение для самых разных сценариев.

Первые годы нашей работы ушли на построение инфраструктуры, анализ бизнес-процессов с формированием первоначального технического задания и саму разработку. Ядром новой системы стал механизм, который был способен рассчитывать цены туров в реальном времени, что позволяло получить преимущество перед конкурентами за счет возможности обновлять цены на путёвки столько раз в день, сколько это необходимо.

В то время отели рассылали ценники массовой рассылкой по электронной почте. Тот туроператор, которому удавалось первым вывесить посчитанные путёвки на своем сайте, получал начальную волну туристов, пользуясь отсутствием конкуренции. С новым функционалом «Библио-Глобус» получал преимущество в часы. Ныне у больших туроператоров разница в появлении новых цен сократилась до нескольких минут, но за эти минуты все готовы биться, потому что они приводят за собой тысячи туристов. Но тогда мы первыми открыли Клондайк!

В соревновании с предыдущей командой разработчиков время расставило всё по своим местам. «Библио-Глобус» функционировал как оперативный штаб МЧС посреди разбушевавшейся стихии. В нём осознанно поддерживалась атмосфера напряженности, чтобы коллектив не расслаблялся. Работа в таких условиях требует стрессоустойчивости, к которой готов далеко не всякий.

И предыдущий штат разработчиков со временем откололся от компании сам. Переход на новую платформу, которую мы разрабатывали, но еще не привели к приемлемому виду, стал необходимостью.

Не всё шло хорошо, система ещё не могла «дышать» самостоятельно, требовала поддержки разработчиками. Руководство теряло веру в успех, но, тем не менее, нам дали возможность продолжить внедрение, что на этом

этапе стало подвигом человеческого духа. Уже после мы задавались вопросом, как оно вообще состоялось? Компания тогда еще была маленькой, разработчики сидели в переговорной (когда она не была занята руководством). Помимо поддержки уже запущенного функционала, требовалось постоянно запускать новые функции, требования к которым тоже менялись. Сотрудники, на себе переживающие тяготы смены системы бронирования, нас тихо ненавидели, а наши соперники не имели желания и мотивации нам помогать.

Помимо бойцовского духа команды, успех проекта стал возможен благодаря собственной технологии управления ресурсами, Мета, уже применявшейся в других проектах и поэтому отлаженной. Используя Мету как платформу для хранения данных и ее передовой (по тем временам) веб-интерфейс, мы смогли дать менеджерам возможность заниматься бронированием еще до того, как разработали все необходимые веб-страницы для отображения информации. Иными словами, мы сумели не прервать бизнес-процесс. Работа компании продолжалась, пока мы разрабатывали недостающий функционал.

В 2008 году Александр Туголуков запустил нашу систему в тестовом режиме. Основные свои направления — Кипр, Таиланд и Тунис — «Библио-Глобус» обслуживал, пользуясь старой системой, а на вновь обретенных — Мальдивы и Индонезия — обкатывал новую.

Нашей окончательной победой можно считать апрель 2010 года, когда под наш контроль передали Таиланд. Эта страна считалась одним из сложных направлений из-за стыковочных перелетов и ценовой специфики, поэтому успешный запуск на ней снял все опасения руководства.

После этого остатки предыдущего ПО доживали свои дни до полной замены новой системой.

Достижение лидерства

Сведя воедино уникальные отраслевые знания и создав функционал расчёта туров в реальном времени, компания «Библио-Глобус» оторвалась от конкурентов на годы. Было очевидно, что, в конце концов, рынок придет к этому же, но такой алгоритм нельзя шаблонно скопировать, он слишком сложен. И это не преувеличение: когда нам понадобилось, мы сами не могли

перенести эту разработку с одной платформы на другую в течение нескольких лет.

Технологическое преимущество позволяло компании первой реагировать на изменения рынка и извлекать из этого дополнительные доходы, которые «подпитывали» наши дальнейшие технологические прорывы.

Все последующие запуски новых больших продуктов, которые влекли за собой шумиху на туристическом рынке, давали нам технологическую фору. Фору не создать, переделав дизайн сайта или добавив SMS-уведомления — через месяц это будет у всех. Фора создается новым продуктом на стыке нескольких направлений или технологий, за счет того, что компания уже владеет этими направлениями по отдельности, плюс платформой, на которой может их успешно совместить. А конкурентам не хватает компетенции или нужных данных, или денег для быстрого создания подобного продукта. При таких условиях можно получить год или два преимущества для доведения до ума нового продукта и реализации следующего.

В 2011 году мы запустили в обиход электронный пакет документов, что было необходимым шагом на пути автоматизации бизнеса. На тот момент в туристической индустрии практиковалась работа с бумажными документами. Мы начали экономить на бумаге, на курьерах и сокращать время доставки информации клиентам. Сегодня все привыкли к получению информации и документов по электронной почте. Но не так давно это было еще в новинку.

В 2009 году «Библио-Глобус» стал стратегическим партнером компании «Трансаэро», а в 2012 году отношения между туроператором и этой авиакомпанией стали настолько близкими, что мы смогли воплотить в жизнь уникальный для отрасли проект системы регистрации на рейсы авиакомпании с удобным графическим интерфейсом. Разработка задумывалась как имиджевый продукт, но оказалась не только «вишенкой на торте», но и маркером преимущества перед другими туроператорами, и впоследствии туристы считали эту бесплатную, но приятную услугу обязательной частью туристического пакета. Наши клиенты в то время составляли значительную часть пассажиропотока «Трансаэро», и в пиковые дни число регистраций через туроператора было сравнимо с числом регистраций на сайте авиакомпании.

Мы пытались развить тему 3D-регистрации и дальше, но большинство проектов не удалось по технологическим, экономическим и другим причинам. Не получилось запустить регистрацию на рейсы и стойки сдачи багажа в гостиницах, мобильную регистрацию на рейс по дороге в аэропорт. Мы разработали терминалы для самостоятельной регистрации в аэропортах, но они требовали постоянной поддержки, доработки и себя не оправдали. Были планы распространить регистрацию на рейсы авиакомпаний «Эмирейтс» и «Катар», но до промышленного запуска дело не дошло. Тем не менее, демонстрационный режим регистрации на множестве красочно оформленных типов самолётов составляет часть технического портфолио туроператора.

В том же, 2012 году «Библио-Глобус» разместил веб-камеры на побережье нескольких отелей Египта, Таиланда и Вьетнама. Во Вьетнаме запустить проект полноценно не получилось: из-за высокой влажности камеры выходили из строя. По стечению обстоятельств, через несколько недель, 12 апреля, случилось землетрясение в Индонезии, и двое суток весь туристический рынок жил новостями из Таиланда, опасаясь цунами. Нагрузка на сайт в это время выросла десятикратно.

На волне такой популярности камер проект должен был продолжиться в Египте. Планировалось закупить порядка семидесяти всепогодных камер, были договоренности с местным муниципалитетом. Мы просчитывали логистику доставки камер, но в последний момент запуск был отменен.

Тем не менее, идея выросла в проект телемоста с отелем. Мы обеспечили средства двусторонней связи, чтобы отдыхающие могли рассказать родственникам на камеру, как у них обстоят дела, а те в ответ написать им текстовое сообщение. Ответы на вопросы могли быть получены в реальном времени, и это снимало с нас упреки в плохом качестве услуг. Подобный сервис имеет смысл только в том случае, если он открыт всем и без предварительной модерации, но довольно скоро мы обнаружили, что наш телемост «обогатился» спамом и нецензурными выражениями Проект пришлось свернуть.

В 2013 году мы запустили наш легендарный механизм динамического ценообразования. Это было бы невозможно без расчёта туров в реальном времени, и новым проектом мы оставляли за бортом всех тех туроператоров, которые не успели нас догнать в области технологии.

Нами была построена многокритериальная модель автоматической регулировки стоимости тура для достижения прибыльности, которая в реальном времени пересчитывала цены, выводящиеся на сайте.

В первоначальной модели цена менялась слишком быстро, и пока агентства вносили фамилии туристов, уже уходила далеко от первоначально показанной на сайте, причем в произвольную сторону. Приходилось замедлять эти изменения.

Рынок, привыкший к статическим или медленным изменениям цен, был возмущен, вплоть до угроз судом или разрывом договоров. А затем конкуренты стали выпускать похожие продукты, анонс подобной системы появился на сайте РЖД, и рынку пришлось под нас подстраиваться.

Тогда же был запущен проект «3D-отель». Для этого проекта была полностью оцифрована кипрская гостиница Amathus Beach Hotel Limassol, на основе чего был создан путеводитель с возможностью забронировать конкретный номер. Для этого было отснято несколько десятков тысяч панорамных фотографий, на основе которых потом строилась навигация по отелю. Проект готовился к MITT (Moscow International TRAVEL & TOURISM Exhibition) и стал подтверждением технологического лидерства компании. Сейчас на нашем сайте представлено в оцифрованном виде около двадцати гостиниц с разным уровнем детализации.

2013 год был богат на проекты. К MITT была запущена интерактивная карта авиарейсов компании «Библио-Глобуса». Мы в реальном времени визуализировали на карте мира десятки рейсов самолетов с нашими туристами, что наглядно подтверждало огромную величину человекопотока, которым управляет туроператор. Эта же карта была интегрирована с системами «Трансаэро» (для правильной отрисовки типа самолёта) и аэропортами (для работы с расписаниями).

В этом же году мы занялись автоматизацией принимающих компаний. Была проведена большая работа по предоставлению им нужного среза информации из системы «Библио-Глобуса» — без дублирования, экспорта и импорта, и при этом удобного в работе. Специально дописывались новые пласты функционала, включая создание единой экскурсионной системы. Насколько мы знаем, «Библио-Глобус» первым на туристическом рынке ввел мобильные принтеры для сотрудников принимающих компаний,

интегрированные с системой бронирования, что позволяло представителю принимающей стороны оформить у себя на планшете экскурсию и выдать билет, оперируя информацией с наших серверов.

Подобная автоматизация была частично вынужденной мерой, когда стали понятны масштабы хищений из кармана туроператора. Даже если туроператор купит принимающую компанию, это не избавит его от серых схем на местах. Только провязывая данные и денежные потоки между всеми участниками бизнес-процесса, мы получаем реальную картину происходящего. То есть, когда продаваемые экскурсии в обязательном порядке вводятся в систему на общих основаниях, а билеты печатаются тоже из нашей системы, тогда в денежные отношения между представителем и туристом вовлекается туроператор. А при наличии первичных данных в системе весь денежный поток становится прозрачен и подконтролен.

Мы акцентировали внимание на технологиях, и каждый новый продукт подтверждал нашу репутацию как технологического лидера. Мы осознанно заявляли о себе не только как о туристической компании, но и как о компании, способной создавать уникальные программные решения.

Следует заметить, что ни один из анонсированных нами проектов не был красивой, но пустой оболочкой. Все эти продукты решали одну из трех задач: уменьшение себестоимости, увеличение оборота или получение технологической форы, поэтому к ним относились весьма серьезно. Все они строились на инфраструктуре «Библио-Глобуса» и использовали уже накопленный функционал и опыт разработки. В свою очередь, наработки, полученные при их создании, потом повторно использовались в других сценариях.

Вместе с известностью пришла и зависть. «Библио-Глобус» благодаря низким ценам на свои туристические продукты давно вызывал недовольство у игроков на рынке: его регулярно называли тайным проектом спецслужб, гигантским должником, финансовой пирамидой и т. д.

Злопыхатели не верили, что все продажи эффективны за счет низкой себестоимости. Но небольшому штату требуется меньший зарплатный фонд, меньше площади под офис и денег на оргтехнику. А поскольку человеческий фактор приводит к тому, что при увеличении нагрузки количество и

стоимость ошибок увеличивается, меньшее число людей, участвующих в процессах создает меньше ошибок, выливающих в реальные деньги. И это тот самый случай, когда автоматизацию можно сделать идеальным исполнителем детально сформулированного процесса, достигая реальной экономии.

Автоматизация бизнес-процесса позволила закладывать минимальную маржу и выигрывать за счет увеличения оборота, а не за счет прибыльности продаж. Однако подозрения в том, что мы чего-то там химичим, не исчезли и по сей день, несмотря на многолетнюю деятельность «Библио-Глобуса» при невысоких ценах и регулярном выпуске высокотехнологичных продуктов.

По счастью, кампанию в СМИ удалось отбить, перехватив повестку, и те публикации, которые попали в прессу, не отражали масштаб первоначальных замыслов. Главной задачей руководства в течение этого трудного времени было сохранение коллектива и репутации «Библио-Глобуса». Над этим очень плотно работали.

В этот период новости о дальнейшей судьбе «Трансаэро» приходили два раза в день. За активы компании боролись несколько сильных игроков, и туристический рынок шатало от этой неопределенности, что не шло на пользу никому. Отрасль активно делала ставки на то, в какой момент обрушится туроператор «Библио-Глобус».

Первой с рынка ушла «Трансаэро», а Александр Туголуков, владелец «Библио-Глобуса», заявил, что как бы много мы ни потеряли, компания будет жить, что бы ни говорили в прессе и как бы ни пытались нас похоронить.

Следует заметить, что когда-то в интервью Forbes Александр Туголуков сказал, что компания способна пережить сокращение в три раза. В сентябре 2015 года она сократилась в 20 раз и выстояла. Не без потерь – закрылись региональные офисы, пришлось отказаться от ряда программ, но компания выполнила все свои обязательства перед туристами.

Кризис 2014

К началу 2014 года «Библио-Глобус» стал крупнейшим российским туристическим оператором. Объемы его продаж выросли в сто раз по

сравнению с 2004 годом. При относительно небольшом числе сотрудников самого туроператора его инфраструктура обслуживала нескольких тысяч одновременных пользователей (без учета клиентов и агрегаторов), представляя собой электронную торговую площадку, пакетирующую в режиме реального времени актуальные предложения авиакомпаний, отелей, экскурсионных компаний и всех остальных наших партнеров.

К тому времени Россия дополнилась Крымом и начались локальные внешнеполитических связях и, как следствие, туризме.

Руководство «Библио-Глобуса», предугадав дальнейшее ухудшение ситуации в отрасли, занялось диверсификацией рисков. Стартовал проект автоматизации бизнес-процессов в отелях. Прогнозируя рост внутреннего туризма, компания начала покупать гостиницы и выкупать гостиничные номера в России. Ухудшение экономического состояния страны и населения привело к снижению объемов летнего туристического сезона и банкротству ряда туроператоров средней руки. Однако вопреки панике на рынке, «Библио-Глобус» сумел улучшить свои показатели, увеличив прибыль и число туристов. В том же, 2014 году, нами был запущен конструктор туров, благодаря которому турист самостоятельно выбирал себе отель и время пребывания в нем. Конструктор дал возможность людям экономить, собирая тур из дешевых отелей. Отели, в свою очередь, получали возможность регулировать свои цены напрямую в нашей системе, управляя продажами.

Валютная паника в декабре 2014 года практически парализовала туристический рынок, подстегнув только продажи горящих туров. Люди скупали электронику и горящие путевки, боясь, что завтра все подорожает вдвое.

Летний сезон 2015 года прошел под знаком роста курса валюты и развития внутреннего туризма. Компания продолжила вкладывать ресурсы в Крым и Сочи, покупая аттракционы, бассейны и прочую курортную Инфраструктуру.

Были запущены автобусные туры, а также выкуплен поезд «Москва-Анапа» с отправлением два раза в неделю. Под новые транспортные средства мы дорабатывали системы регистрации, рисовали 3D-модели и продумывали алгоритмы рассадки пассажиров. Мы не сразу поняли специфику железнодорожных туров: здесь путешествовало много маленьких детей с

бабушками, а для них верхние места в вагонах были крайне неудобны. Поэтому первоначальную автоматическую рассадку пришлось отключить, особенно после того, как автоматика от безвыходности начала отсаживать маленьких детей в другие купе. Было принято решение обслуживать клиентов по схеме «при покупке тура сразу выберите места», чтобы туристы еще до оплаты тура знали, как они будут располагаться в вагоне.

В это же время в компании появились аппаратные направления, сначала как НИОКР, но, чем сложнее обстояли дела на туристическом рынке, тем больше внимания уделялось собственному производству.

В тот же летний сезон «Трансаэро» устроила гигантскую распродажу, демпингуя на большинстве направлений и существенно увеличив пассажиропоток. «Библио-Глобус», как основной клиент этой авиакомпании, смог удешевить туры и начал отправлять в августе по 20 тысяч человек в день, несмотря на ослабление рубля и общую обстановку в стране. Это позволило компании по итогам девяти месяцев вырасти на пять с половиной процентов, при общем провале рынка на десятки процентов.

Но с начала сентября у «Трансаэро» возникли проблемы с обслуживанием долгов, и правительство мгновенно ввело временное управление, при этом управляющим был назначен «Аэрофлот» — прямой конкурент «Трансаэро». Туристический рынок запаниковал — до конца сезона на самолетах «Трансаэро» должно было улететь еще полмиллиона туристов. Продажи туров резко упали.

Через две недели временная администрация «Трансаэро» запретила продажи авиабилетов. К этому моменту «Библио-Глобусом» уже была предоплачена программа полетов до конца декабря на 5 млрд рублей. Остановка продаж основного перевозчика мгновенно сократила объем предложений на сайте до туров на регулярных перелетах и вынудила оперативно искать замену.

За месяц с начала сентября ежедневный оборот «Библио-Глобуса» сократился в 20 раз, но компания сохранила операционную деятельность, хотя подобное падение пережить удастся немногим. Даже в разы меньшие падения ломают компании, и то, что «Библио-Глобус» удержался — это значимый урок для рынка. К примеру, печально известный в отрасли туроператор «Лабиринт» имел оборот в 20 раз меньше, чем «Библио-Глобус»,

а персонала в 4 раза больше. У нашей компании не было такой избыточности при несравненно большей эластичности, что и позволило ей выдержать удар.

Наблюдая развитие ситуации в сентябре от плохого к худшему, конкуренты старались воспользоваться пошатнувшимся положением «Библио-Глобуса» и готовили масштабную кампанию по его дискредитации в СМИ. Уход крупнейшего туроператора вместе со второй по величине авиакомпанией России освободил бы место для остальных участников рынка — на плотном, перенаселенном рынке важно не быть первым, а не уйти первым. Авиакомпании могли получить более свободное небо, а туристические компании — наших туристов.

Жизненно важным было сохранить доверие агентств, чтобы поддержать поток продаж. При утрате доверия оборот упал бы до нуля и компанию можно было закрывать. Поэтому были предприняты следующие шаги:

- 1. Всем агентствам разрешили не платить за туры до вылета.**
- 2. Всем, кто захотел, вернули деньги за их туры.**

Это добавило агентствам уверенности, что они в любом случае не потеряют деньги, и обеспечило непересыхающий ручеек продаж, который поддержал компанию в течение осени.

Из-за ангажированности СМИ в новостях складывалось впечатление, что «Библио-Глобус» доживает свои последние дни, но это не соответствовало действительности. Компания, построенная на лучших в отрасли технологических процессах, оставалась управляемой, необремененной долгами и раздутым штатом и продолжала развиваться. Уменьшилось число самолетов и туристов, но это не повлияло на бизнес-процессы.

16 октября Александр Туголуков дал большое интервью РБК, где четко выразил уверенность в том, что на туристическом рынке будущее за теми, кто сумеет грамотно автоматизировать бизнес-процессы и использовать передовые информационные технологии. Вот отрывки из этого интервью:

— (РБК) Вы наверняка слышали мнение ваших конкурентов о «Библио-Глобусе»: единственное его конкурентное преимущество — это цена, но, как он получает такие цены от «Трансаэро», загадка.

– (АТ) Я помню, что начиная с 2000–2001 годов всегда «Библио- Глобус» упрекали в цене, в демпинге: «Откуда они эти цены берут? Как они работают?»

Поэтому в какой-то момент времени я перестал на это реагировать. Ведь людям гораздо проще мыслить шаблонно. Ведь никто в деталях разбираться не хочет. Потому что в динамическом ценообразовании, которое реализовано было у нас, все построено на минимальной стоимости. Мозг реагирует на минимальную цену. Но какой объем путевок по этой минимальной цене, никто не знает. А стоимость разную глубину продаж разная. И гораздо проще обвинять «Библио-Глобус», чем признать то, что технологически мы сильнее, чем наши конкуренты.

Но я как был убежден, так и сейчас остаюсь убежденным, что побеждает технология. Если сейчас я решу, что целесообразно законтрактовать пул других перевозчиков и перенести туда наши возможности, ничего не изменится. Я это уже проходил. Просто технологии – это не только определение стоимости. Технологии – это минимизация действий человека, т.е. когда все изменения, которые проходят в системе, делаются роботами, а не людьми – вот что такое технология. Никто же не говорит о том, что у меня сотрудников в несколько раз меньше, чем у компаний, которые гораздо меньше меня по объему. Это самая важная вещь, уникальность в этой части.

...

– (РБК) Ваши коллеги утверждают: если они работают в минус (к примеру, по Таиланду или Кипру и прежде всего из-за того, что «Библио- Глобус» ведет себя очень агрессивно), то не может сам «Библио- Глобус» не терять деньги.

Ваш туроператорский бизнес прибылен?

– (АТ) Если бы мы не были прибыльными, если бы мы были в убытке, каким образом я бы мог продолжать бизнес?

– (РБК) Поэтому и предполагают: кто-то за «Библио- Глобусом» стоит, кто-то его финансирует.

– (АТ) Поверьте, не требуется этого. Никого вы никогда не найдете, чтобы кто-то вам принес деньги. Это невозможно, это надо забыть. Если ты вкладываешься во что-то, это даст тебе успех и принесет рентабельность и все прочее. Но в это надо вкладываться. Не просто пользоваться чем-то стандартно. Если вы готовы разбираться в вопросах, что-то создавать новое, тогда оно и придет. Но, поверьте, в реальности таких людей очень мало. В основном люди хотят чем-то пользоваться, а создавать не хотят.

...

(АТ) Туроператорский бизнес работает сам по себе. Денежные средства поступают, механизмы отточены, они работают эффективно. Перед собой я сейчас ставлю задачу реализовать софт для отелей, потому что отельный софт здесь неэффективный, он очень дорогой. Я для этого взял в аренду отели в России в этом году, чтобы начать это готовить, и планирую продолжать это дальше. Это гораздо интереснее — выйти с софтом и разместить его в тысячах гостиниц! Потому что в авиации, в отелях в основном софт иностранный, то есть все хранилище там. Отрубить его — это мгновение. И это помимо того, что вся информация располагается в базах данных, в том числе и наших оппонентов в мировой политике. Это ненормально. И я четко понимаю, что мы эту вещь можем изменить. Я в этом вижу колоссальные перспективы.

Если бы мы сейчас сделали софт для отелей, софт для авиакомпаний — жизнь прожита не зря. Поверьте мне, я вижу только в этом интерес для себя. Удовлетворение от денег мне получить сложно: я не умею отдыхать. Поэтому от реализации каких-то вещей я получаю удовлетворение, тем более что это еще и необходимо.

...

— (РБК) В самом начале разговора вы сказали, что непременно выполните все обязательства перед своими клиентами. Но почему тогда у всех ваших компаний, которые продают туры, финансовые гарантии меньше 100 млн руб.? У ООО «БГ Карибы», через которое идет весь турпоток за границу, это всего 30 млн руб. Это же ничтожная сумма при ваших объемах.

— (АТ) Очень важно понять, зачем мне здесь гарантии? Для чего они нужны?

— (РБК) Мало ли, случится землетрясение или еще какой-нибудь форс-мажор и «Библио-Глобус» станет финансово несостоятельным.

— (АТ) Не нужны. Не нужны эти гарантии. Вы взяли денежные средства, вы не можете оказать услугу — вы их вернете. Вы же взяли!

Через две недели после этого интервью произошел теракт на борту А321 в Египте, и 8 ноября был наложен запрет на авиасообщение между странами. Это стало вторым ударом по туристической индустрии, поскольку Египет является основным зимним пляжным направлением.

«Библио-Глобус» это событие не подкосило, но пошатнуло. Тем не менее, компания продолжила деятельность, переориентируя своих клиентов на Кипр. Рынок, которому тоже не поздоровилось, вышел на новый виток ставок на закрытие нашей компании.

А 24 ноября после инцидента в Турции Ростуризм закрыл и турецкие направления. Больше всего это ударило по крупным туроператорам с турецкими корнями, для которых Турция была их вотчиной. В нашей компании к этой новости отнеслись спокойно и продолжали работать, ведь Турция – не зимнее направление и поток туристов «Библио-Глобуса» туда невелик. Дополнительно против турецких туроператоров сыграл тот факт, что в после закрытия Египта они переводили своих туристов на Турцию. Наши основные конкуренты получили свою встряску и начали съезживаться.

Дешевых популярных направлений не осталось, пассажиропотоки сократились, у авиакомпаний возник избыток мест, цены на самолеты в Сочи обвалились, но мгновенно выросли цены на сочинские отели. Численность туристов упала до минимума, и это происходило на всем рынке, а не на отдельных направлениях.

31 декабря на волне антитурецких санкций из реестра туроператоров были исключены компании с турецкими корнями, являющиеся прямыми конкурентами «Библио-Глобуса»: Coral, Pegas Touristic, Anex tour, Sunmar. Все они входили в десятку крупнейших в стране, терять рынок не планировали, поэтому в январе вернулись обратно, сменив собственников или юрлица.

К весне 2016 года стало понятно, что Египет будет закрыт по меньшей мере на весь летний сезон, и прогнозы говорили об общем сокращении турпотока на 30% по сравнению с предыдущим годом.

Восстановление

В конце лета 2016 года в прайс-листах туроператоров снова появилась Турция. Туристический сезон уже подходил к концу, но, тем не менее, открытие этого направления вызвало определенный ажиотаж. За этот год рынок серьёзно переориентировался на внутренний туризм. Были запущены экономичные туры на поездах РЖД, «Библио-Глобус» продолжил

инвестировать в отели Крыма и Сочи, выкупая их полностью или частично.

Спустя два года от разработки ПО для автоматизации бизнес-процессов гостиниц мы перешли к внедрению, интегрируясь в информационную систему компании.

В 2016 году Туголуковы взяли в управление крупнейший в Европе трехзвездный сочинский отель Azimut, переименовали его в «Сочи Парк Отель» и запустили туда наш ИТ-отдел для автоматизации разрозненных бизнес-процессов в этом гостиничном гиганте (почти 3 тысячи номеров).

Для поддержки нового бренда в переходный период, стоимость проживания была снижена на 90%. Это не было актом отчаяния или предбанкротными судорогами. Продажа номеров существенно ниже себестоимости имела вполне практический смысл: насытить отель туристами, готовыми при такой дешевизне простить огрехи новой информационной системы, и, одновременно, заявить рынку о новом игроке. Добавим к этому перенасыщенное самолетами сочинское направление, и сформированный тур оказался доступен всем желающим. Распродажа закончилась на майских праздниках 2017 года, когда сервис был отлажен, и загрузка вышла на требуемые уровни.

Параллельно, в 2017 году торговый дом «Библио-Глобус» (то есть крупнейший в Европе книжный магазин на Лубянской площади), оценив эффективность информационной системы туроператора, согласился передать управление своими данными нашему ИТ-отделу. После нескольких месяцев работы по миграции данных и перестройке процессов, мы в начале лета запустили новый интернет-магазин. Синергия позволяет нам использовать этот ресурс для продажи дополнительных услуг в турах, например, «книга в дорогу».

Глава 2. Прикладная часть

В 2004 году, когда Игорь Антонов договаривался с Александром Туголуковым о разработке новой информационной системы «Библио-Глобуса», он обозначил следующие условия для работы своей команды:

1. Достаточно высокие — выше рыночных — зарплаты, самостоятельный набор людей в команду, самостоятельные технологические решения.
2. Право на ошибку. У ИТ-отдела есть некий условный бюджет, в рамках которого могут быть допущены ошибки.
3. Работа с первым лицом, и только с ним. Задачи отделу может ставить только наниматель, все остальные запросы на доработки проходят только через него.

Такой подход снимает очень многие конфликтные ситуации и не позволяет разным отделам получать преференции в ущерб работе компании. В тот момент, когда первое лицо теряет интерес к разработке и делегирует принятие решений подчиненным, все начинает разваливаться.

Технология

Согласовав условия и принимаясь за работу, мы понимали, что быстрого результата не будет. Разработка новой системы «Библио-Глобуса» должна была иметь свой базис. Нельзя взять и начать писать, например, страницу бронирования туров. Для этого нужна платформа, и не как идеологическое обоснование, а как набор готовой инфраструктуры и компонентов, уже интегрированных между собой и готовых к работе с данными. К этому относится и отлаженный API низкого уровня для работы с данными, сообщениями или прочим, что понадобится на уровень выше.

Позволить себе автоматизировать бизнес-процессы без платформы могут только компании в крайне медленно меняющихся отраслях. Там, где все меняется часто и помногу, только грамотно построенная платформа дает основу для безболезненной поддержки изменений. Тех, кто приходит с энтузиазмом, но без платформы, скорее всего, задавят.

Первый год автоматизации деятельности «Библио-Глобуса» ушел на построение инфраструктуры. Заключались договоры, строились сети, закупались серверы и площадки в дата-центрах, настраивались БД, то есть формировалась основа для дальнейшего роста. Система с самого начала строилась для работы с несколькими дата-центрами, репликацией и отказоустойчивостью (это не было добавлено после!) и впоследствии могла практически безболезненно расширяться в десятки раз по данным и функционалу. Потом, при возникновении необходимости, такие качественные скачки в архитектуре становятся слишком дорогими и задерживают развитие бизнеса.

Будучи фундаментом для автоматизации, интерфейсы (API) нашей платформы становились уровнем абстракции, скрывающим от программистов всю глубину системы. Новый программист, начинающий писать на API платформы, использует всю мощь инфраструктуры, но не замечает этого. Ему кажется, что он пишет богоподобные вещи, но на самом деле его собственных разработок — проценты, все остальное позволила сделать платформа, в ней это уже заложено. Чем мощнее становилась платформа, тем быстрее она позволяла разработчикам создавать новый функционал, строить и перестраивать бизнес-процессы. А за счет скорости разработки решение получалось дешевле, благодаря чему компания могла обгонять рынок и оставаться в выигрыше.

Большой ошибкой стало бы начинать разработку бизнес-логики, имея только представление, как будет связана инфраструктура. Начинать стоило, только построив платформу и проверив, что все действительно работает. После формирования платформы программная (а иногда и аппаратная) ее часть является основой, которую крайне сложно заменить. На практике это ведет к фиксации интерфейсов, привязке к конкретным БД и замороженным архитектурным решениям. Это не так плохо до тех пор, пока вы не вынуждены по какой-то причине заменить архитектуру системы. В нашем случае это был отказ от проприетарной СУБД в пользу открытой. Аппаратная часть становится проблемой в тот момент, когда необходимость изменения архитектуры упирается в ограничения технологий вендора. Мы столкнулись с этим в книжном магазине «Библио-Глобус», где специализированное вендорное хранилище

(и связанная с ним архитектура системы) не позволяло разнести данные на несколько дата-центров.

Информационную систему «Библио-Глобуса», создаваемую на базе платформы, мы стремились сделать информационно провязанной, но энергетически разорванной. Она должна была иметь возможность передавать информацию потребителям во все свои уголки. Реализовать эту возможность можно по-разному, самая интуитивно понятная — это шина данных. А показателем правильной реализации становится способность локализовать проблему и ограничить ее распространение. Если с какой-то частью системы произошла беда — программная или аппаратная, — то остальные должны узнать об этом, но продолжать действовать независимо, иначе отказы перестанут быть локальными и начнут поражать разные части системы. Другими словами, мозг всегда знает, что делают пальцы, но волдыри появляются только в месте ожога.

В этой книге мы намеренно не приводим описания аппаратного и технологического стека «Библио-Глобуса». Для нас подбор подходящих технологий базируется на других принципах. Для начала — те сторонние технологии, которые применяются в «Библио-Глобусе», уже устарели, как и устарели все популярные сегодня технологии, поскольку с момента их рождения, обкатки, популяризации и распространения прошло два-три года, и сейчас люди пользуются ПО, которое пора уже, по-хорошему, обновлять.

Все стараются строить промышленные решения на вещах, проверенных временем и десятками других внедрений. Наша идеология основана на том, что если компания хочет быть первой, она не должна делать как все. Ни один из тех технологических гигантов, кто сейчас задает стандарты индустрии — Apple, Google, Facebook или подобные, не делал как все, и до сих пор не делает. Технологические решения у больших компаний рождаются свои. Скорее известна обратная зависимость — если большая компания начинает поддерживать какую-то технологию, вкладывает в нее ресурсы и дает обратную связь, то эта технология не исчезает в неизвестность¹.

¹ Читателя может заинтересовать сайт techstacks.io, на котором приводятся списки используемых в популярных компаниях технологических стеков. Там достаточно материала, чтобы понять, что общего у всех остальных, и чем отличаются лучшие. И сделать по-своему.

При создании своей технологии разумно выбирать самый простой и правильный путь для решения нужной задачи. А после этого оптимизировать решение – оптимизировать так, чтобы сделать его самым производительным в индустрии. Как, например, SAS Institute, который сделал самое быстрое хранилище для больших данных. Или Андрей Аксенов со Sphinx – он просто на каждом этапе добавлял два-три процента производительности. Понятно, что чем ниже уровень оптимизируемого кода, тем больший выигрыш дают эти проценты.

Если же разработчик опирается на чужие технологии, то они должны быть не самыми популярными, а самыми удобными для решения требуемых задач. Результат, безусловно, окажется субъективным – можно долго спорить, какая технология лучше подходит под ту или иную задачу.

В конечном счете, время покажет, насколько правильным был выбор. При этом есть ненулевая вероятность, что технология не оправдает себя и ее придется заменить. Поэтому в «Библио-Глобусе» помимо жесткого отбора кандидатов на новую технологию используемый функционал принудительно ограничивается базовым набором функций. Если есть возможность написать эквивалент полезной, но экзотической функции с использованием более простых функций – этот путь предпочтительней, поскольку он не так болезненно переживает смену технологии. Сложные и уникальные функции приковывают разработчиков к технологии и ограничивают развитие системы в дальнейшем. Мы предпочитаем не строить впереди себя эту стену. В практике работы «Библио-Глобуса» была технология хранения данных, которая имела супербыструю репликацию. После перехода на неё и запуска в промышленное использование, оказалось, что при большом количестве серверов репликация остается супербыстрой, но теряется 15% данных. Технологию пришлось оперативно заменять.

Еще одно ограничение выбираемых технологий – их способность масштабироваться. Будет правильным предполагать, что создаваемая вами система растянется на весь мир. Это условие требует использовать подходящие решения и правильным образом. Работать в одном дата-центре и работать в нескольких – это принципиальнейшая разница. «Вконтакте» не смогла растянуть данные на несколько дата-центров (ДЦ), и, наверное, уже не растянется. Мы изначально решили, что система «Библио-Глобуса» будет ставиться на несколько ДЦ, что этот порог придется перейти, и сразу решали

инфраструктурные и технологические проблемы с этим связанные. Потому что потом, когда пройдет несколько лет и нам надо будет иметь несколько ДЦ — а это обязательно будет в большой компании, — мы не должны столкнуться с проблемами. Система изначально получилась сложнее, чем могла бы быть, но зато ее развитие не было заторможено качественными скачками инфраструктуры.

Технологии и инфраструктура должны быть подготовлены к работе в режиме 24x7. Если обновление продукта возможно только ночью, пока никто не работает, то это потенциальная проблема, потому что однажды придет момент, когда продуктом будут пользоваться круглосуточно. Другими словами, есть отрасли, в которых остановка процессов для обновления и обслуживания может привести к проблемам и простоям бизнеса.

Считать, что информационная система вырастет до глобальных масштабов — не мечтательство. Так ведь и будет — иначе зачем все затевать? И поэтому разумно сразу выбирать (или создавать) технологии, которые не будут препятствовать росту.

Наша работа на упреждение проблем роста продолжалась всё время. Нам поверили потому, что архитектурные и инфраструктурные решения позволили компании очень быстро вырасти, не упираясь в лимиты системы. Но для этого каждый год делался четырехкратный запас ресурсов, и каждый год половина этого запаса выбиралась благодаря росту компании. Двукратный запас оставался, и его хватало на перепады нагрузки. А летом 2015 на информационную систему компании обрушилась трехкратная нагрузка, и она уперлась в потолок, поскольку такой запас мощности не закладывался и такая нагрузка не ожидалась.

Практические выводы — система всегда должна иметь двукратный запас ресурса. Больше — дорого, дорогие серверы используются неэффективно. А меньше — отсутствует запас на скачки нагрузки, которые бывают, например, на праздничные дни или при каких-то внешних колебаниях рынка.

Опыт показывает, что на практике запаса ресурсов не будет. Если система может выдерживать пиковые нагрузки, то никто не будет отключать серверы в спокойное время — скорее, сервисы будут предоставляться с меньшей задержкой или в большем объеме. А если что-то отдавать быстрее или больше, то рынок это все заберет.

Кажется, что если тратить на страницу поиска не секунду, а триста миллисекунд, то будет доступно в три раза больше мощностей. Это не так, потому что потребители рассуждают: «Ага, за триста. Тогда я запрошу не 50 результатов, а 500». Они довольны и быстро привыкают к хорошему, но стоит чуть «просесть» в период нагрузок или переконфигурации, как их собственные системы начинают работать медленнее, вызывая проблемы и разочарование.

Поэтому в каждый момент времени нам нужно иметь возможность увеличить мощности. Мы держим четырехкратный запас и уже жизнью научены, что на все свободные ресурсы сразу возникают и потребители этих ресурсов. А сам запас — не только аппаратные мощности, но и возможность балансировать нагрузку, и возможность хранить и распределять данные, то есть программная составляющая платформы.

Автоматизация

Компании, зарабатывающие внедрением систем управления предприятием или процессами, вкладывают в этот термин следующий смысл: человеку за компьютером надо дать возможность решать свои задачи быстро и удобно. В «Библио-Глобусе» нами была применена каноническая трактовка:

автоматизация — это когда на рутинных операциях человек становится не нужен. Таким образом, получается жесткая, негуманная, но эффективно работающая схема — люди остаются только там, где требуется творческий подход к делу или взаимодействие с другими людьми. Но последствия её применения могут быть разными: сокращение персонала либо его перевод на участие в бизнес-процессах следующего уровня сложности, либо расширение операционной деятельности компании, либо какой-то компромисс между этими вариантами.

Чтобы автоматизация проходила успешно, руководство обязано эту схему понимать и принимать. В нашем случае, при разработке системы каждый бизнес-процесс разбирался с руководителями компании с этих позиций, что и привело к такому выигрышу в результате.

Но при внедрении нужно обязательно помнить о целесообразности. При

переходе от теории к практике мы старались держать в голове, зачем автоматизация нужна. Как правило, это две цели.

1. Уменьшение затрат.
2. Уменьшение количества ошибок.

Например, подошла очередь автоматизировать бизнес-процесс, которым занимается сотрудник на зарплате. Для понимания, стоит ли вообще это делать или лучше оставить все как есть, прибегнем к такому алгоритму:

1. Зарплату сотрудника сопоставляем со стоимостью автоматизации и поддержки (зарплата сотрудников ИТ, расходы на инфраструктуру).
2. Оцениваем, как изменится количество и критичность ошибок.
3. Оцениваем, какие ситуации и бизнес-процессы, связанные с этим сотрудником не могут быть автоматизированы за разумное время и деньги, их частоту и критичность.

Если ИТ-отдел играет на стороне работодателя, то он учитывает, что автоматизация — не вещь в себе и не способ освоения бюджета. Она призвана в первую очередь уменьшать общие затраты компании и количество ошибок. Нарушение этого правила вынуждает бизнес тратить деньги на самих автоматизаторов вместо развития.

В нашем случае другого сценария быть не могло: «Библио-Глобус» переживал период ежегодного двукратного роста в течение десяти лет при малом числе персонала и настолько быстро вышел на серьезные объемы, что автоматизация была единственным способом справиться с ситуацией. В противном случае сотрудники генерировали бы такое количество ошибок, что это не смогло бы удержать компанию на плаву.

Чтобы не уходить далеко от темы, рассмотрим еще ситуацию, когда разработчик пишет веб-страничку для менеджера. Он старается, делает красивые кнопки и выпадающие меню и при этом получает зарплату в несколько раз больше рядового сотрудника. С первого взгляда ничего плохого не происходит, но эта автоматизация нерентабельна, по сути, разработчик шантажирует компанию — поскольку не уменьшает затрат.

Менеджер все равно не перестанет работать и будет получать зарплату. Но ситуация станет другой, если разработчик пишет страницу, которой станут пользоваться пятьдесят менеджеров, тридцать из которых потом можно

будет уволить. Сработает правило: либо сокращение расходов, либо увеличение оборота. Ведь если не достигается ни одна из целей автоматизации, то она бесполезна. В таком случае получается, что деньги высасываются у компании без прибавления функционала, и число таких случаев следует минимизировать.

Даже при описанных выше требованиях, есть допустимая ситуация, при которой автоматизация может быть нерентабельной. Это имиджевые и исследовательские проекты. Формирование образа высокотехнологичной компании требует существенных вложений, но отзывается повышением стоимости на рынке. Известно, что компании начинают стоить в десятки и сотни раз больше при том же функционале, если их назвать ИТ-компаниями. И это оправдано: более дорогая компания имеет бонусы, люди сознательно и несознательно стремятся работать с лучшими на рынке.

Для этого, конечно, надо действовать на опережение, выпускать что-то такое, что еще никто в отрасли не делал. Таким имиджевым ИТ-проектом для «Библио-Глобуса» стал, например, уже упомянутый «3D-отель», в который было вложено очень много денег, но он заставил рынок (и не только) по-другому оценивать компанию. После него мы уже не могли отойти в сторону и тихо работать дальше — подобные проекты должны выходить регулярно, и это вынуждает нас постоянно придумывать, чем еще удивить рынок.

Кстати, ИТ-компания — это необязательно ИТ-услуги. Это имя. Приписка бренда, который ИТ. Если речь идет не о туристической, а об ИТ-компании по продаже туров, то весьма вероятно, что она будет стоить дороже.

Мы начали автоматизацию, и у нее сразу же появились противники. Главный противник во все времена — это лень, а у менеджеров еще и страх. Страх сидит в людях, которые привыкли работать так, как работали. Человек, услышав, что вот сейчас его труд будут автоматизировать, начинает подозревать, что, скорее всего, после этого его уволят, и начинает сопротивляться.

Бороться с этим можно двумя способами. Можно, конечно, давлением, но эффективнее закольцовывать бизнес-процессы так, чтобы сотрудник начал зависеть от работы другого сотрудника, и, в свою очередь, плоды его работы использовались кем-то дальше по цепочке. При такой взаимосвязи человек уже не может не делать (либо делать плохо) свою часть работы, у него уже не получится сопротивляться автоматизации. Все важные данные оказываются

использованы, а ненужная шелуха отпадает сама. И только тогда процесс начинает работать.

Отрицательный пример из жизни: сотрудник заносит данные договоров в систему. Но эти данные нигде не используются, для них не выстроена обратная связь, договор можно вбить или нет — ни один процесс от этого не зависит. Сотруднику могут приказать вбивать договоры, но у него всегда найдётся более важное дело, и в результате ничего вноситься не будет. Такой подход не работает.

В разработке то же самое — разработчик строит свои процессы на чьих-то других. И все, что он делает, пускается в ход, от его наработок начинают зависеть другие процессы и люди, и это дает обязательную обратную связь. Теперь разработчик должен поддерживать свой код в хорошем состоянии, он быстрее узнает об ошибках и быстрее их исправляет. К примеру, разработчик пишет программу, которая ранжирует текст по шкале «хорошо-плохо» для оценки отзывов на отели. Просто показывать эту оценку около отзыва — мертвая информация. Но если начать её использовать для поднятия отеля в результатах поисковой выдачи, то образуется кольцо обратной связи. Очень скоро отель начнет возмущаться, почему он оказывается в хвосте списка, разработчик начнёт разбираться, выяснит, что у него слово «олдскульный» имеет негативный окрас, и исправит свой алгоритм. Не замкнув обратную связь, нельзя побороть естественную лень человека и заставить разработчика отладить свой алгоритм до совершенства.

Независимо от того, разработчик это или менеджер, всё, что он делает, должно замыкаться на общий процесс и иметь обратную связь. Но у разработчика есть еще одно ограничение — все его разработки не должны занимать много времени. Если задача реализовывается слишком долго, то она становится ненужной и от неё отказываются, а это прямые потери денег и времени.

Лень дает еще один эффект — разработчики подсознательно стараются решать только привычные задачи или заниматься привычной им поддержкой. Но для развития информационной системы и самой компании этот путь неэффективен. Приоритетными должны стать самые сложные задачи, которые позволят компании получить фору. Если у разработчика есть

несколько задач, и какая-то из них «выстрелит» — следует решать её, остальные задачи либо потеряют актуальность со временем, либо «созреют» и станут более важными.

Такой подход неизбежно ведет к накоплению технического долга, описанного ниже, в главе 4, и задача «разбор накопленного» со временем повышает свой приоритет до уровня, когда этим действительно следует заняться.

Для роста компании нужно контролировать ИТ-отдел и понимать, что все выдающееся сейчас разрабатывает именно он.

Три шага

Примененный нами подход к автоматизации описывается следующим алгоритмом.

1. Нужно автоматизировать определенный уровень иерархии в компании, например, авиаотдел.

Существует уровень выше, который ставит нам эту задачу (в «Библио-Глобусе» задачу ставят только владельцы), и есть уровень ниже — который предоставляет данные для работы. Есть также несколько отделов того же уровня, взаимодействующие с авиаотделом и между собой.

Понятно, что уровень, с которым идет работа, не должен сам ставить задачи на разработку, так как правильной автоматизации в этом случае не получится. Тот же авиаотдел не может поставить задачу так, чтобы в результате его сотрудников сократили, это нонсенс. Он может ставить задачи, которые касаются лишь облегчения их работы, но это не принципиальное решение вопроса. Принципиальное — снижение трудозатрат или человеческих ошибок.

2. Сначала мы анализируем, от чего этот уровень зависит информационно — его входы, выходы, соседей. Для выписки билетов сотрудники авиаотдела взаимодействуют с системами бронирования авиабилетов (уровень снизу), а для сохранения информации — с отделом бронирования (тот же уровень, соседний отдел). Наш опыт показывает, что результат никого не устроит, если до конца неизвестно, что внизу и по бокам, и как оно работает. Не зная досконально весь процесс передачи и обработки данных, можно даже не

браться за эту задачу.

3. После изучения процессов мы, используя средства автоматизации, начинаем замыкать входы с выходами. Сначала всё что нужно замыкается на соседей, потом на уровень выше. Поэтому первым шагом бронировщики (соседний или верхний отдел) замыкаются на систему бронирования (уровень снизу), получая возможность выписывать билеты самостоятельно. Для уровня выше нужны данные по продажам, поэтому мы замыкаем на руководство первичные данные из систем бронирования, получая и формируя необходимую статистику и нужные отчеты. В обратную сторону идёт поток управления, и руководство получает контроль над настройками, которые раньше обрабатывались автоматизируемым отделом. Это может быть глубина выписки билетов, величина наценки или список разрешенных систем бронирования. Таким образом, от функций отдела откусывается по кусочку, пока тот не перестает быть нужен вообще или в текущем объеме.

В идеале, в компании все должно работать автоматически. Люди должны заниматься творчеством, а не рутинной работой. И творчество для них – это решение нестандартных задач. «Библио-Глобус» смог быстро вырасти, поскольку здесь стандартные задачи отданы автоматике. Менеджеры занимаются тем, что мы не успели автоматизировать. Все что могло быть формализовано, в основном было формализовано и реализовано. Но для этого нам приходилось преодолевать сопротивление, потому что люди сами никогда не скажут, как нужно правильно построить процесс, ликвидирующий их основную рутинную работу и оставляющий только творчество. Они будут смотреть со своей точки зрения, а на задачу всегда надо смотреть с более высокого уровня.

Мы поднимались на уровень выше, понимали что нужно сделать, досконально изучали нижележащие процессы и смыкали верхний и нижний уровень. Так успешно решались задачи автоматизации.

Как уже упоминалось, при этом мы не забывали о целесообразности. В жизни существуют сценарии, когда для замены менеджера ИТ-команда силами троих разработчиков потратит в 100 раз больше ресурсов, и потом еще будет поддерживать процессы. Экономика такой автоматизации оказывается совсем неэкономной, но, по счастью, это можно выявить на этапе анализа. Человек очень многие вещи делает лучше и эффективнее автоматике, особенно вещи нестандартные, потому что нестандартные вещи делать интереснее.

Экспансия

Компания росла быстрее рынка, и в какой-то момент ее начала тормозить нетехнологичность смежных отраслей. Туризм включает в себя разноплановых контрагентов, начиная от банков и заканчивая музеями, и контролировать потоки данных в такой гетерогенной среде очень трудно. Например, если платеж проходит полдня, то вы не сможете продать тур за два часа до вылета и потеряете деньги; если экскурсовод оперирует наличными — он всегда найдет способ вас обмануть.

По мере роста компании и ее влияния на туристический рынок, очень серьезные усилия были направлены на поддержку интеграции с внешними источниками. Сначала на условиях контрагентов, последние годы — на наших. Мы смогли добиться, чтобы в околореальном времени в нашу систему попадали данные из платежных систем, страховых компаний, банков, принимающих и экскурсионных компаний. Время, потраченное на разработку подобного механизма, окупается увеличением оборота и блокированием серых схем работы с деньгами. Начинается такая экспансия, как функционал перекачки данных между нами и системой контрагента. Это быстрое, но некрасивое решение, и со временем мы дорабатываем нашу платформу для непосредственной работы прямо в ней, избавляясь от сторонних продуктов и необходимости поддержки интеграции. Тут мы убиваем сразу нескольких зайцев.

Прежде всего, автоматизировать замкнутую систему, в которой хранятся все первичные данные, легче — вы сами являетесь агентами изменений. В реальности по мере роста компании и выхода на новые рынки приходится поддерживать интеграцию со многими внешними системами. Ситуация, когда ваш бизнес-процесс зависит от функционирования неконтролируемой системы — перманентная головная боль, усиливающаяся с ростом количества стыковок. Мы регулярно сталкиваемся с проблемами доступности сервисов, которые перекладываются на плечи клиентов и вызывают совершенно оправданные негативные отклики. Приходится прибегать к запасным сценариям, но, к сожалению, всех неожиданностей заранее предсказать нельзя.

Важно понимать, что перебои чужих сервисов не должны остановить ваш главный бизнес-процесс; если так происходит, то архитектура системы нуждается в исправлениях. «Библио-Глобус» старается иметь альтернативный доступ к каждому критически-важному сервису или альтернативный функционал, реализующий то же самое. В нашей истории случались периоды длительной недоступности ключевых партнерских сервисов, когда спасало только наличие запасных путей. Например, при работе с «Трансаэро» мы не могли полагаться только на веб-сервисы системы бронирования, а вынуждены были автоматизировать эмуляцию терминала, предназначенного для бронировщиков — потому что он работал через другой канал связи. Это в буквальном смысле нас спасало несколько раз. Время, затраченное на разработку альтернативы, окупается при отказах многократно. К тому же это не дает посредникам выкручивать вам руки, торгуя доступом к сервису. Хуже, чем перебои — неожиданные изменения спецификации вендорами. Случаи, когда вендор предупреждает об изменениях, можно пересчитать по пальцам, и торговые площадки могут отвалиться в самый неподходящий момент. Опыт показывает, что потери можно минимизировать только оперативной реакцией и доработками, что недостижимо при использовании аутсорс-разработки, посредников, или агрегаторов услуг. К нам периодически обращаются поставщики решений, предлагающие автоматизировать тот или иной аспект нашей деятельности. Ни один из них не может показать ту способность изменять свое решение под изменяющиеся требования и скорость реакции, которая требуется для сохранения скорости роста. При этом стоимость решений будет сопоставима со стоимостью найма разработчика в штат, а временные затраты — сравнимыми.

На рынке присутствует много агрегаторов услуг, например, при продаже авиабилетов. Они торгуют едиными API доступа к разным системам, освобождая клиента от необходимости самостоятельно разрабатывать и поддерживать «зоопарк» стыковок.

Это заманчивый рынок, но он не оправдывает себя на больших объемах. Затраты на разработку стыковки покрываются в первый год использования сервиса. Использование посредника требует разработки такой же стыковки, но с ограниченными возможностями и невозможностью повторного использования. Собственная разработка дает свободу.

Вторым преимуществом собственных решений в смежных отраслях становится финансовая прозрачность контрагентов. При выходе на новые

рынки и сферы индустрии мы неизбежно начинаем взаимодействовать с компаниями-партнерами, представляющими интересы региональных участников. Все они понимают, чем грозит переход на нашу систему. Они теряют независимость, начинают вести все данные и деньги у нас, и их нежелание может быть пересилено только их заинтересованностью. Наше решение должно быть лучшим на рынке, позволяющим нашим партнерам экономить за счёт автоматизации процессов и минимизации персонала.

К слову, это относится и к региональным филиалам, дочерним компаниям и подобным структурам. Когда из центрального офиса им говорят: «Теперь вы будете работать вот так», они чаще всего отвечают: «Да что вы там видите, я с этим каждый день вожусь! Это не заработает». На самом деле они видят не больше. Они уже зашорены, а со стороны лучше видно, что можно сделать. Конечно, улучшения заработают только в том случае, если разработчики досконально разобрались в бизнес-процессах. Только когда автоматизаторы понимают, как и почему компания работает именно так, а не иначе, они могут выстроить процессы оптимально.

Но это промежуточный шаг. Нашей конечной целью является выход на непосредственных поставщиков услуг и автоматизация их деятельности. В туризме партнер — это компания, которая имеет хорошие отношения с гостиницами, и это всё, что имеет значение. Если убрать прослойку партнера и выйти напрямую на автоматизацию стыковки с гостиницами, то получится выигрыш в эффективности, скорости и управляемости, причем как для компании, так и для гостиницы. Гостиницы тоже конкурируют друг с другом, и если предложить им технологическое решение, дающее преимущество, это может оказаться весомым аргументом. Мы смогли предложить такой функционал в 2016 году.

Помимо партнеров, туристами занимаются и принимающие компании, и туристические агентства, и индустрия развлечений, и все они получили или ещё получают своё отраслевое решение, интегрированное в общую систему. «Библио-Глобус» проводит экспансию, поглощая и автоматизируя небольших представителей этих сфер, а затем предлагая свои решения всему рынку. В принципе, «Библио-Глобус» — это тоже посредник между гостиницами и туристами, только очень крупный и владеющий технологиями. Заводя на себя потоки данных и договоры от десятков тысяч контрагентов, компании такого масштаба создают весь туристический рынок, которым пользуются все

остальные. Они, в отличие от посредников, продают продукт, которым владеют, и поэтому несут ответственность за туристов. Чтобы лучше контролировать ситуацию, туроператоры пытаются развиваться в обе стороны — от поставщиков услуг, с одной стороны, до покупателей туров — с другой, пробиваясь через посредников и заменяя модель B2B на B2C.

Развиваясь до такого масштаба, туроператоры сами становятся объектами спроса других, более мелких компаний. Мы снова были обязаны разрабатывать API, на этот раз для доступа к своим ресурсам со стороны других сервисов. Нашим API сейчас пользуются тысячи компаний-посредников, добавляя какие-то свои услуги, агрегируя информацию с рынка и находя свои способы добавить стоимость продукту. Количество запросов от автоматизированных систем растет лавинообразно, в разы превышая число наших пользователей на веб-сайте. Когда «Яндекс» делал свой проект поиска туров первый раз, его роботы приходили к «Библио-Глобусу» в таком количестве, что серверы выходили из строя от нагрузок. Но это тоже экспансия. Тот, кто владеет API (и данными) — контролирует потребителей, и наша задача — сделать наш API самым востребованным на рынке.

Безопасность

При разработке мы делали акцент на скорость, корректную обработку данных, а безопасности уделялось меньше внимания, чем хотелось бы. Но по мере роста системы рос и интерес к ней, и в какой-то момент нас начали проверять на прочность. Для защиты от несанкционированного доступа применялись дополнительные контуры безопасности — поскольку принципиально невозможно создать одноуровневую защиту, спасающую всегда и от всего. Мы реализовали многоуровневое решение с пониманием того, что может случиться плохого, если каждый уровень попадет в чужие руки. Чем серьезнее информация, тем больше контуров нужно пройти для доступа к ней, и это касается как внешнего мира, так и собственных пользователей системы.

«Библио-Глобус» работает со множеством партнеров из разных стран, предоставляя им свои данные или храня чужие и тем самым пропуская их сквозь свой внешний периметр. Никто не может гарантировать, что своими правами доступа не воспользуются недобросовестные партнеры или хакеры

из-за слабой безопасности партнеров, которую нам сложно контролировать.

В свою очередь, нельзя доверять и собственным пользователям системы настолько, чтобы разрешать им больше необходимого. Обычных сотрудников не особо заботят проблемы безопасности. В свое время по компании проходили рейды со сдиранием с мониторов бумажек с паролями.

Поэтому доступ к данным, словно пирог, нарезался кусками до тех пор, пока каждый не получил ровно тот кусочек, который нужен ему для работы. Такой подход при разумном применении уменьшает последствия и от инсайда, и от торговли информацией — угроз, свойственных высококонкурентным рынкам. Плюс, он спасает от обычной людской глупости и безалаберности — случайный пользователь не сможет открыть ссылку, опубликованную в переписке, а даже если откроет, то не увидит в ней чувствительных для компании данных и тем более не сможет их изменить.

Но доступ к данным — это частный случай. В первую очередь атакующие пытаются получить доступ через сетевую инфраструктуру или вызвать её отказ. Последние годы у компании не проходило дня без попытки обрушить серверы или получить несанкционированный доступ к системе. Характерно, что число и интенсивность DDoS-атак очень сильно коррелируют со значимостью происходящих в туристической индустрии событий.

Защищаясь от внешнего мира, компания вынуждена защищать внешний периметр буквально от всего, начиная, безусловно, с целенаправленных атак, до ошибок пытающихся стыковаться внешних систем или слишком агрессивных агрегаторов. Большие поисковые компании вроде «Яндекса» или «Гугла», приходя за информацией, своими запросами вполне могут сломать неподготовленную инфраструктуру.

При этом используемые для работы с информационной системой «Библио-Глобуса» клиентские сервисы не позволяют обойтись только внешним периметром — внутренних пользователей тоже нужно принудительно разводить «по своим углам» на уровне сетевой инфраструктуры. К примеру, зарубежный экскурсовод не должен физически получить доступ к чему-либо, кроме экскурсий на своей, специально выделенной веб-странице.

Времена, когда компания была стартапом и каждый мог делать практически все, прошли. «Библио-Глобус» за счет автоматизации бизнес-процессов и усложнения доступа к ним постоянно повышает планку надежности, отнимая у пользователей доступ к исходным данным. Все меньше сотрудников могут увидеть ключевые данные в системе и повлиять на них.

Глава 3. Роботы работают

Правила

Запустить новую систему в компании — это рассказать всем про новые правила игры и заставить по ним играть. Правила, конечно, постепенно меняются, поскольку развитие не останавливается. Но именно принятие правил сотрудниками и разработчиками является важным фактором успеха.

Прямым следствием автоматизации «Библио-Глобуса» стало правило отказа от рутинных операций. Люди начали понимать, что простые вещи за них надёжней делают роботы, и в какой-то момент приняли это за данность. Когда сотрудники билетного отдела жалуются, что им приходится выписывать билеты — это успех правильного подхода к автоматизации рутины.

Вторым, таким же важным, стало правило персональной ответственности.

Базовым свойством построенной системы является механизм хранения истории всех данных, проходящих через неё. При необходимости можно получить слепок данных на нужный момент времени, что, помимо повышения энтропии во вселенной, имеет прикладной аспект: всегда можно восстановить, кто менял данные и каким образом. История данных используется в случае порчи или утери какой-то информации, а сотрудники понимают, что им не удастся свалить проблему с данными на роботов или кого-то другого.

Помимо сохранения данных, сохраняется и история запросов к веб-страницам, дополняющая картину — можно понять, какие кнопки нажимал сотрудник и по каким страницам ходил, чтобы привести данные в такое состояние. Разумеется, это все относится и к внешним пользователям системы.

Использование этих инструментов позволяет идентифицировать и наказать конкретного сотрудника. Но одновременно каждому предоставляется контроль над ситуацией и результатами действий в своей зоне ответственности — через отчеты и уведомления по своим процессам. Однако иногда писем становится слишком много, а отчеты слишком большие, и контролировать ситуацию сотрудник не в состоянии, а может только

реагировать на проблемы.

Для реализации правила отказа от рутинных операций создаются многочисленные автономные кусочки функционала под общим названием «роботы». Запускаясь по расписанию или по событиям, они автономно выполняют поставленные задачи. Это и обслуживание самой системы, и функционал перенятых у человека процессов – общим числом их уже около тысячи.

Экспоненциальный рост нагрузки и сложности системы привел к тому, что сейчас в большинстве процессов уже нет возможности остановить роботов и вернуть управление человеку. Для этого потребовалось бы многократно увеличить штат. Даже временная недоступность системы начинает быть ночным кошмаром

разработчиков и ИТ-директора. На пике высокого сезона минута простоя выливается в тысячи долларов упущенной выгоды.

Но это не единственная беда. В некоторых процессах вынужденное бездействие должно быть наверстано к определенному моменту, например, формирование документов при получении оплаты. Отыгрыш упущенного времени тоже требует времени, и далее рекурсивно. Технически возможно рассчитать величину простоя как с участием человека, так и без него, после которого нагнать упущенные операции становится невозможным – а это означает необслуженных туристов, суды и, как следствие, потери.

Обратная связь приходит разработчикам как от роботов, так и от сотрудников. Идеал, к которому мы стараемся приблизиться – робот должен сам распознать некорректную ситуацию (как недоступность внутренних или внешних сервисов, так и некорректные входные данные) и сам сделать все возможное для исправления или, в крайнем случае, предупреждения человека. Здесь хорошо зарекомендовал себя принцип Парето – приложив 20% усилий к доработке автоматике, мы закрываем 80% возможных проблем. Риски снижаются на какое-то время, но дальнейший рост компании довольно скоро раздувает оставшиеся 20% процентов до первоначальных масштабов, что приводит к необходимости новой итерации.

Раньше сотрудник выполнял неавтоматизированный процесс один раз в день, теперь десять, скоро это будет занимать все его время. Такие данные собирают (сотрудники охотно присылают их сами), им расставляют

приоритеты, выясняют, не преувеличен ли масштаб проблемы. Иногда оказывается, что или проблема надумана, или этот процесс достаточно изменить, чтобы проблема ушла. Есть ситуации, когда обсуждение автоматизации необходимо эскалировать до уровня руководства — в случаях, когда сотрудники хотят автоматизировать процесс в частности, в таком виде, к которому привыкли, вместо построения оптимизированного решения.

После того, как задача сформулирована, от разработчика требуется решить её максимально качественно в буквальном смысле. Он должен сделать и забыть, иначе затраты на поддержку съедят всю экономию от автоматизации. Например, разработчик не вник в задачу и сделал её плохо (или как ему проще), а потом начинает ругаться с менеджером и дорабатывать свой результат под уточненные требования. Но время этого разработчика стоит в пять раз дороже времени человека, чью деятельность он автоматизировал. Случается, иногда переделывать некорректно решенную задачу оказывается дольше и дороже самого решения.

Обновление уже работающих процессов должно быть подвержено такому же правилу — каждая итерация должна быть такой, чтобы результат работал без непредусмотренных побочных эффектов. Если вдруг разработчик уйдет в отпуск на два месяца посередине переделки процесса — никому не должно стать хуже.

Поддержка начинается, когда разработчик выполнил задачу небезупречно, потому что не разобрался досконально. Имеет смысл использовать дополнительное время на вникание в процесс, а не на приближение результата к требованиям.

Однако если выполнять задачу безупречно, то на это понадобится намного больше времени. При этом ценность задачи со временем уменьшается, и вопрос «насколько безупречно делать» оказывается многокритериальным. Качественное, но задерживающееся решение принесет меньше пользы компании в тактическом плане. Решения, которые готовятся годами, не приносят отрыва от конкурентов. Следует учитывать, что само понятие «безупречно» становится все более расплывчатым, если подниматься по лестнице иерархии управления. На более высоком уровне мелкие огрехи в процессах не видны, в отличие от уровня разработчика или сотрудника

технической поддержки.

У ИТ-компании нет шансов выжить и прокормить себя, если разработчики занимаются поддержкой своих задач. Рентабельность процесса разработки при этом начинает стремиться к нулю.

Подчеркнем, что мы относим понятие безупречности исключительно к бизнес-процессам в компании. Оно не относится к удобству и красоте веб-страницы, которую видит всего один сотрудник. Его страница должна просто и надежно работать, и главное — что с этой некрасивой страницы он формирует продукт, которым пользуются миллионы пользователей сайта.

Скорость, с которой страница сайта открывается у клиента, важнее скорости, с которой она разрабатывалась, потому что окупается многократно. Функционал для нас важнее удобства разработки и простоты тестирования. Тут критерием правильности разработки является круглосуточная работоспособность основного бизнес-процесса компании. По понятным причинам — он обеспечивает оборот денег. Для туроператора это, например, поиск и продажа туров с сайта, приём денег и выдача документов. Любой простой основного процесса, помимо потери денег, вызывает резко отрицательные эмоции со стороны всех участников — от менеджмента до потребителей, — и серьезные репутационные потери на рынке. В больших и очень больших компаниях это наносит еще и политический урон.

Никакое обновление кода или технологии, никогда не должно останавливать основной процесс. Согласно закону Мерфи что-то обязательно может пойти не так, и лучше лишний раз протестировать критические участки процесса, чем тратить нервы на исправление обнаруженной на сайте компании ошибки.

Наверное, читателю уже стало понятно, что после безупречности следующий ключевой критерий в ИТ — это время. Время важнее денег, деньги можно занять. Время не займешь, всё должно делаться для его минимизации. Всё должно быть быстрее: быстрее работать система, быстрее проводиться разработка. Мы вынуждены упрощать сложные вещи и оптимизировать медленные. Постепенно нагрузки вынуждают нас переходить от неэффективных реализаций алгоритмов к эффективным. Это, кстати, не совпадает с путем многих больших компаний, которые вместо оптимизации предпочитают просто увеличивать серверные мощности до достижения желаемых показателей и нанимать обслуживающий штат.

И еще один критерий, определяющий разработку, — это гибкость производимого продукта. Для бизнеса она важнее функциональности, поскольку позволяет запускать новые функции быстрее, пусть даже в ограниченном объёме. Разрабатываемый функционал становится намного качественнее только после попадания в кольцо обратной связи. Да, обратная связь может быть очень неприятна, но чем быстрее мы её получаем, тем меньше лишнего делаем и быстрее запускаем процесс. В частности, поэтому для нас является ошибкой ставить красоту реализации на первое место в ущерб функциональности и работоспособности. В литературе у такого подхода есть много разных имен. К примеру, у Алистера Кокбурна (Alistair Cockburn) он именуется Walking Skeleton (<http://alistair.cockburn.us/Walking+skeleton>), и он действительно эффективен в agile-разработке¹.

Гибкость позволяет бизнесу развиваться. Нас просят о новом функционале, мы его быстро делаем и пускаем в работу. Пусть получится неуклюже, проблемно для выкладки, но все сто тысяч человек, которые зайдут на сайт сегодня, получают что-то новое, и это важнее, чем красота реализации и удобство разработки.

В отделе есть печальный пример разработки, продолжавшейся три года без промышленного запуска. Разработчик получал зарплату за продукт, имеющий все шансы быть устаревшим к моменту запуска, потому что он ставил приоритетом качество кода, эффективность использования ресурсов и прочие вещи, которые будут бессмысленны в случае провала проекта. Обратная связь не замкнулась, и неизвестно, что на самом деле было важным в продукте. Разработчика уволили, а продукт не запустили.

ИТ-отдел

Первоначально разработчики сидели в общем офисном пространстве с менеджерами, так что каждый, проходя, мог зайти поругаться или рассказать про новую задачу. Получалось и неудобно, и неформализовано, и весьма отвлекало от дела. Поэтому в 2011 году в отделе появился специально обученный сотрудник технической поддержки, через которого стало

¹ Гибкая методология разработки (англ. Agile software development, agile-методы) — серия подходов к разработке программного обеспечения, ориентированных на использование итеративной разработки, динамическое формирование требований и обеспечение их реализации в результате постоянного взаимодействия внутри самоорганизующихся рабочих групп, состоящих из специалистов различного профиля. (Wikipedia)

проходить большинство звонков и писем с проблемами.

Это был продуктивный шаг, но отдел стал черным ящиком, периодически выдающим новые продукты. Сотрудники не понимали, чем занимаются эти высокооплачиваемые люди за стеной, что, конечно, не добавляло хороших отношений. Это же привело к деперсонализации во взаимодействии между ИТ и всеми остальными. Конечно, сотрудники знали, кому именно звонить в случае проблем, но у отдела общая репутация и общая ответственность.

Изначально разработка новых продуктов велась циклически: в межсезонье – разработка, в сезон – поддержка. Доля времени на поддержку процессов в высокий сезон доходила до 80%. Но затем рост компании, как и связанный уровень нагрузки, перестал уменьшаться в межсезонье, и стало неважно, когда вы запускаете новый продукт, а когда идете в отпуск. В 2015 году высокий сезон наступил весной, и так и не закончился, потому что правила игры на туристическом рынке менялись с неприятной регулярностью.

По мере роста компании в команду приходили разные люди. Не все уживались, далеко не все были готовы к ритму разработки. Основными качествами, отличающими подходящего для «Библио-Глобуса» разработчика, можно назвать следующие:

1. Заинтересованность и вовлеченность в процесс. Нужно понимать, что ты делаешь, зачем, и какие побочные эффекты возможны у этого бизнес-процесса. Отстраненность не дает построить правильное решение. Позиция “я только исполнитель” приводит к череде дальнейших переделок из-за недостаточного вникания в процесс.

2. Ответственность. Одно из ключевых качеств, когда человек держит руку на пульсе своей зоны ответственности, а в идеале и смежных. Это включает способность решать внезапные проблемы посреди ночи, в баре с друзьями и прочих ситуациях. Равно как и способность принять ответственность на себя.

3. Быстрая реакция. На больших объемах любая ошибка всплывает в течении нескольких минут. Ошибка в неудачном месте может стоить тысяч долларов в минуту. Поэтому способность почувствовать, что что-то не так, понять, что именно не так, исправить без ошибок и быстро выложить исправление очень ценится.

4. Стрессоустойчивость. По мере погружения в тему, с ростом количества внедренных процессов, ростом ответственности, растет нагрузка на каждого. Особо нагрузка увеличивается в высокий сезон, когда число обращений на

поддержку вырастает в разы. Сотрудник с неустойчивой психикой может неадекватно среагировать на увеличение нагрузки, и поэтому ненадёжен.

5. Адекватность. Умение сопоставлять свои желания с возможностями и опытом — ценное качество. Статистика говорит о том, что отсутствие перегибов и здоровое понимание вещей присуще далеко не всем.

6. Психологическая сочетаемость. В коллективе при сложных внешних условиях сложилась своя атмосфера, и непопадание в нее, естественно, вызывает отторжение.

Опыт собеседований показывает, что перечисленные качества в совокупности встречаются на рынке весьма редко. Возможно, поэтому большинство сотрудников отдела пришли по рекомендации коллег.

Кстати, перечисленные качества ничего не говорят о знании языков программирования или теоретической подготовке кандидата. В первую очередь потому, что самого кандидата крайне трудно найти. Научить намного легче. За полгода (раньше ему все равно не дадут ответственную часть) человек способен разобраться с языком программирования, базовыми вещами и решить несколько маленьких задач. В отделе до последнего времени была стандартная задача для новичков — внести улучшения в редактор данных отдела контента. За последние годы через этот редактор прошли практически все новички, попутно разобравшись в инфраструктуре системы и принятых в компании подходах к разработке. И, если функционал всё-таки кто-то ломает, есть и другие способы обновить данные. Это не критичный функционал — он не влияет основной бизнес-процесс.

В больших компаниях и корпорациях штат и процесс разработки организуются из принципа взаимозаменяемости людей. Это ведет к полному покрытию кода тестами, ревью кода, детальной документации и прочим особенностям современных методологий разработки.

Но вспомним, что Google и Facebook начинали с подвоя гаражной разработки и только потом были вынуждены превратиться в корпорации. Такой “творческий” подход применяется и в «Библио-Глобусе». Да, люди заменяемы. Но следующий человек переписывает то, что ему не понравилось в наследии от предыдущего или в чем он не разобрался. Это оказывается эффективным, поскольку обычно существуют проблемы с небольшим участком системы и нет надобности единомоментно выкорчевывать сразу всё.

В «Библио-Глобусе» минимальная текучка кадров и ситуации с унаследованным кодом разовые. Этому способствуют ответственность разработчиков и интересные задачи, которые им приходится решать. Единый процесс тестирования был развернут только в середине 2014 года, а документирование так и не состоялось как неприоритетная задача, и сейчас это личное дело разработчиков.

Смысл был в том, что на тесты и документацию тратится время — основной ресурс программистов. Лучше это время потратить на разработку системы и создание уникальных качеств, которые дадут отрыв от конкурентов. Но при растущей сложности системы недотестированный код начинал вызывать все больше проблем, и процесс разработки был принудительно изменен добавлением обязательного контроля качества третьим лицом.

Вообще, упор на минимизацию времени относится не только к автоматизации, но и ко всем процессам в компании. Непроданное место сгорает, скорость нужна, чтобы его продать сегодня, а не завтра.

Поэтому критически важна скорость принятия решений. Быстро принятые и реализованные решения быстрее приведут к ошибке или успеху — и следующей итерации.

Отсутствие документации и покрывающих тестов не приводят к ущербу в качестве, если придерживаться нескольких важных принципов:

1. Не делать изменений, которые разрушают целостность системы.

Изменения структуры состоят из нескольких этапов. К примеру, нам нужно сменить хранилище данных (подход применим и к переделке бизнес-процессов).

- Параллельно с текущим разворачивается новое хранилище.
- Мы начинаем писать данные сразу в оба хранилища.
- Потом создается конвертер, который заберет все данные из старого хранилища и положит в новое.
- Затем чтение данных постепенно переводится на новое хранилище.
- И, наконец, останавливается запись в старое хранилище.

Такая схема позволяет в любой момент остановить или откатить процесс без ущерба. Не будет ситуации, когда данные разбиты по двум системам и их не собрать. Появляется возможность мигрировать постепенно, с обратной связью на каждом этапе, а это необходимо для систем, которые нельзя останавливать. На предприятиях, где можно всё остановить, изменения проводятся намного проще (правда, с ненулевой вероятностью потом не запуститься). Но следует закладывать на то, что наступит момент, когда останавливать систему уже будет нельзя.

2. Чем больше организация, тем меньше изменение. Любое изменение способно сломать процесс, а в сложной системе это сделать намного проще. Логично, что атомарная переделка должна касаться только одного процесса, и хорошо, если это возможно. Компенсацией малого объема изменений является их количество.

3. Чем больше кода, тем он должен работать быстрее. Очевидный вывод, базирующийся на том, что сложный код содержит много уровней абстракции и большую глубину вызовов. Этот тезис не означает, что небольшие системы имеют право работать медленно, а акцентирует необходимость применения интеллекта при разработке систем, рассчитанных на большие масштабы.

С другой стороны, дополнительные уровни абстракции делают разработку удобнее и проще, уменьшая количество ошибок. Поддержание правильного баланса скорости и удобства часто выражено функцией от скорости работы. Другими словами, начинает работать медленно — оптимизируем в ущерб понятности.

В компании не любят сторонние библиотеки — их надежность и скорость не гарантированы. Большинство разработчиков библиотек не оптимизируют свой продукт под действительно высокие нагрузки и низкое потребление памяти. Более того, в библиотеках возможны проблемы с безопасностью, и такие прецеденты случались. В 2015 году исследователи обнаружили дыру в безопасности, связанную с популярной открытой библиотекой `commons-collections`. Настолько популярной, что она входит в состав многих серверных `java`-приложений, весьма серьезных по своим задачам.

4. Клиентская часть приложения никогда не должна работать с большими объемами данных. Тоже очевидный вывод, связанный с объемом данных и отставанием клиентских мощностей от серверных. Разница может быть не

видна для небольших приложений, но с ростом системы и числа пользователей геометрически увеличивается объем передаваемых данных, что плохо отражается на всех участниках. А когда данные переданы, не каждый клиент сможет их обработать — кто-то работает с телефона, а кто-то с бабушкиного компьютера.

Например, разработчик хочет сделать список всех вариантов автодополнения для формы поиска, но у него несколько десятков тысяч записей. Передавать их при каждом открытии страницы — увеличивать время ожидания на секунды. К тому же, работа самого автодополнения тоже займет секунды при таком объёме. Пользователь не получит удовольствия от работы с такой страницей.

5. У разработчика должно быть больше одной задачи. Всегда должен быть выбор, чтобы сотрудник мог взять задачу, подходящую ему под настроение и состояние. Хорошо, если у него несколько задач. Плохо, если у него несколько десятков задач с одинаковым приоритетом. Существует определенный психологический предел по количеству задач, после которого у человека просто опускаются руки. Лучше, если в таких случаях руководителем будут выставлены приоритеты. Декомпозиция больших задач на маленькие — также дело руководителя, но в любом случае задача должна быть «на вырост», на развитие профессиональных возможностей человека, то есть сложнее и масштабнее задачи в общепринятом понимании.

Задача руководителя — пристально следить за разработкой, чтобы люди не ушли каждый в свою сторону. Поскольку бизнес-процессы сложны, сотрудник может не видеть весь путь, который ему нужно пройти для выполнения задачи. Поэтому новые большие задачи делятся на части, и непосвященным дается объяснение, что делать и в каком порядке — просто потому, что люди раньше этого не делали. Такая работа требует усилий, и руководитель может поддаться слабости самому взяться за кодирование вещей, которые кажутся ему простыми, вместо объяснения задачи подчиненным. Действительно, результат можно получить быстрее, но это менее эффективно для компании, поскольку руководитель использует своё время на менее квалифицированную работу, а сотрудник не проходит часть своего пути в решении задачи.

Одна из задач руководителя — держать разработчиков в тонусе, не давать команде расслабиться и завалить разработку продукта. История знает достаточно примеров, когда разработчики успокаивались, и компания теряла

лидерство на рынке. Большая команда каждый месяц потребляет серьезные деньги, и поэтому обязана постоянно что-то выпускать. От руководителя постоянно требуется генерировать задачи, подстегивать людей, не давать им зарываться в поддержку и мелкие задачи в ущерб выпуску продукта. И тогда получается делать действительно большие и сложные вещи в разумные сроки.

По мере роста системы руководитель теряет контроль над процессами более низкого уровня. Компенсируя это, сотрудники должны забирать этот контроль в своей зоне ответственности, а желательно и в смежных.

Понятно, что разработчики любят заниматься поддержкой своего кода, поскольку он им знаком и задача не требует больших усилий. Но человек должен каждый день делать что-то от большой задачи. И почасовки, которые разработчик составляет, действуют не только как журнал и планировщик, но и как психологический инструмент, напоминающий о сделанном и должном.

Все эти годы модель разработки и управления разработкой менялась, притиралась и обкатывалась. Однажды мы попробовали отдать разработку на аутсорс.

«Библио-Глобус», будучи сильным технологически, не имел штатных дизайнеров интерфейса, и в какой-то момент для повышения удобства сайта редизайн страниц был отдан на аутсорс дизайн-бюро, оно же в дальнейшем строило фронтенд для макетов, нарисованных руками нескольких своих сотрудников. После двух лет сотрудничества мы отказались от подобной практики по следующим соображениям:

1. Чтобы работать в темпе и с нагрузками «Библио-Глобуса», надо быть частью команды. Человек из другой команды имеет другую философию разработки и опыт, поэтому создает трудновстраиваемый и сложноподдерживаемый код. Общая философия — это важно, команды не должны пытаться перетягивать одеяло требуемого функционала на себя, а эффективно решать только свою часть задачи. Например, баланс между функционалом клиентской и серверной частей можно выставить в весьма широких пределах, и неважно, на чём вы остановитесь, важно, чтобы разработчики и той, и другой половины одинаково это поняли и приняли.

2. Система слишком быстро меняется, поэтому нет возможности формализовать ТЗ, и процесс разработки выходит из-под контроля. Отсутствие ТЗ и незнание специфики бизнеса вкупе с удаленной работой приводят к ненужным дополнительным итерациям. Сторонние разработчики,

в свою очередь, работают на почасовой ставке и заинтересованы (сознательно или нет) растянуть процесс. У нас были веб-страницы, разрабатываемые месяцами.

3. Полученный продукт крайне сложно поддерживать собственными силами, поскольку он сложен и использует современные, но неизвестные остальным технологии. Иногда мы получали технологические спагетти, причем сырые, поскольку чужие разработчики за наш счет изучали новые технологии.

Для компании, ценящей гибкость и скорость изменений, это фатальные недостатки. Со временем мы были вынуждены самостоятельно освоить стек современных клиентских технологий, при этом обязательным условием была поставлена одинаковая компетенция минимум нескольких сотрудников – чтобы снова не допустить технологического «зоопарка» и проблем с поддержкой кода.

Сотрудники

Тем, кто собирается повторить путь «Библио-Глобуса», следует сразу понимать, что сотрудники никогда не полюбят автоматизаторов всем сердцем, если от последних зависит рабочий процесс. Программисты пришли, чтобы переделать всё, к чему привыкли менеджеры, чтобы заставить менеджеров работать по-новому, чтобы уволить кого-то (многих) за ненужностью. Иногда программисты что-то ломают. Зачастую отнимают у менеджеров возможность решить проблему, ужесточая права доступа или убирая нужные кнопки.

Поэтому отношение сотрудников к ИТ-отделу ранжируется от терпимости до ненависти. Это приводит к проблемам в постановке задач и скандалам вроде «если вы такие умные, то и делайте мою работу». Если сотрудник заказывает доработку, требуется очень хорошо разбираться в предмете, чтобы отличить, хочет ли он улучшить процесс или просто облегчить себе жизнь. Когда доработку заказывает руководство, оно и является первой точкой для анализа процессов. Руководство может ответить на вопрос «почему», а за вопросом «как» придется уже идти к подчиненным. И можно сразу ожидать, что легко не будет. Некоторых нюансов в специализированных процессах мы добивались от сотрудников годами.

Генерировать правильную обратную связь могут не все. Есть менеджеры, у которых либо все нормально, либо «ничего не работает!». У них сложно добиться конструктива с правильно расставленными приоритетами. А есть люди, понимающие, что новый функционал неполон, но с ним можно построить главный бизнес-процесс, то есть для запуска достаточно. Такие могут выстроить разработчикам необходимый порядок изменений и умеренно терпеливо дожидаться их реализации. Эти люди незаменимы при запуске новых внутренних продуктов или выходе на новые рынки.

Каким бы образом ни была получена постановка задачи, реализация всегда будет отличаться от того, на что рассчитывают сотрудники. Либо из-за переосмысления процесса во время его анализа, либо из-за того, что знания не были переданы полностью или скрыты, либо из-за коррекции процесса руководством. А знания будут скрываться, потому что разработчики в буквальном смысле отнимают у людей хлеб, передавая роботам их зону ответственности.

В любом случае, руководство обязательно получит жалобы на кривизну автоматизации, ошибки роботов, потери денег и уход клиентов. Мы справились с этим только при поддержке владельцев компании, разбирая с ними все проблемные случаи.

Скорость, с которой вносятся изменения в систему, впечатляет рынок, но приводит к проблемам внутри компании. Появляются и исчезают кнопки на страницах, но о них знают только те, кто заказывал функционал. В таких случаях единственным хранителем знаний является ИТ-отдел, точнее, тот сотрудник, который решал задачу. А объем работ у ИТ такой, что делиться этой информацией нет возможности. Документировать систему бессмысленно, поскольку она слишком быстро меняется. Остается стараться делать универсальные решения, которые пригодятся сразу более чем одному человеку и могут повторно использоваться в дальнейшем. И, конечно, стараться выпрямлять внешне и внутренне нелогичные решения для приведения проекта к архитектурной простоте.

Проблема с передачей знаний, вкупе с сохранением истории действий в системе, привела к тому, что сотрудники боятся нажимать на новые кнопки — и, пожалуй, правильно делают. В какой-то момент мы поняли, что передача знаний не работает между отделами, даже если те сидят в одном помещении.

А вот по каким причинам менеджеры не делятся информацией между собой — осталось неясным.

Но существует другая проблема из категории «все девочки так делают», когда непонятные мотивы заставляют как клиентов, так и менеджеров нажимать на неизвестные им кнопки и, что хуже, учить этому других. К примеру, в один из дней 2013 года был запущен функционал самостоятельной аннуляции заявок агентствами. В течение нескольких минут после запуска в техподдержку позвонило семь агентств, которые из любопытства нажали на кнопку и получили аннуляцию со всеми вытекающими штрафами.

Хуже всего бывает сотрудникам, если прежняя кнопка вдруг начинает вести себя по-другому. Это только добавляет негативного отношения к разработчикам. Лучше сделать рядом еще одну кнопку, а потом убрать старую, обеспечив психологический переход, чем поменять (не исправить, а именно поменять) функционал существующей. Еще, конечно, важно правильно называть эти кнопки, чтобы лишний раз не вводить в заблуждение.

Но проблема с передачей знаний существует и внутри ИТ. Объем и история проекта таковы, что никто не знает его целиком, мало кто помнит, кто что делал и тем более как оно работает. Разработчики стараются идти проторенными путями, дублируя собственный и чужой функционал для каждой новой задачи — если лень искать уже готовые наработки по всему проекту. Пожалуй, каждый в своей жизни начинал проект с типовых шагов: обработка строк, обработка дат, работа с файлами, и только потом реализация бизнес-логики на построенном фундаменте. Эта же логика, многократно повторенная, замусоривает проект весьма серьезным образом. Также плохо, что повторно не используется уже реализованная бизнес-логика — контроль доступа, управление ресурсами и т.п. Проблемы такого рода можно исправить, только акцентируя на них внимание и передавая знания о своих разработках коллективу.

В нашем проекте применим тезис **«Каждый разработчик в своё время пытался написать авторизацию»**.

Глава 4. Метаморфоз

Технический долг

Как уже упоминалось, кризис осени 2015 года сильно ударил по компании, поставив её на грань выживания. Операционная деятельность практически прекратилась, оборот уменьшился в 20 раз, всё замерло.

Это был идеальный момент для возврата технического долга и превращения ИТ-отдела в ИТ-компанию.

Первые три-четыре года после запуска платформы в «Библио-Глобусе» о техническом долге никто не вспоминал. Первоначальная архитектура допускала существенный рост, и вся разработка была нацелена на быстрый запуск системы. Мы конкурировали с командой, уже имеющей свое решение, нам надо было её догнать, перегнать, сделать все лучше, гибче и быстрее. Поэтому вначале мы просто работали.

Потом, когда с помощью системы, разработанной за условно маленькие деньги, получилось решение, которое стоило миллионы и которое стало лучшим в мире по оценке сторонних людей — нам поверили. А уже потом, когда команде стали верить, стало возможным отдавать технический долг. И важен он стал далеко не сразу, изначально тратить на него ресурсы было нецелесообразно. Но в какой-то момент мы были обязаны омолодить платформу, чтобы расти дальше. Без этого компания осталась бы на тех объемах, которые уже смогла переварить. Да, эти объемы были велики. Но если у компании есть желание или необходимость растянуть систему на всю страну, то технический долг придётся вернуть. Мы пытались работать с техническим долгом и ранее, но постоянный рост продаж, повышающиеся нагрузки и увеличивающиеся требования бизнеса не позволяли приступить к этому процессу.

Термин «технический долг» был введен Каннингемом в 1992, но он касается деградации систем при их усложнении. Мы же будем говорить о другой стороне долга, связанной с деградацией систем при их масштабировании. Этот долг точно так же приходится возвращать, хотя это не для всех очевидный процесс. Его крайне сложно объяснить финансистам, потому что

он затратен и не приносит очевидной прибыли. В случае, когда компанией управляет совет директоров, согласование бюджета на возврат долга может оказаться проблемой, которая будет решаться по остаточному принципу.

Хорошая, годная платформа не должна ограничивать компанию. Но, в принципе, не должна и ограничиваться компанией. Если платформа может вырасти на целый континент, почему бы ей этого не позволить?

Часть времени ИТ-отдел стал отводить на переделку и оптимизацию. Это не влияло на основные процессы, но было так же необходимо: например, смена хранилища, запуск системы кеширования, оптимизация самой платформы, повышение безопасности или переделка потоков данных. Казалось бы, все это годами работало и лучше в эти дебри лишней раз палкой не тыкать, но мы применяли усилия, а компания тратила существенные ресурсы, подбирая эти долги для того, чтобы впоследствии у нее не было ограничений.

Иногда в процессе обновления система «ломалась», потому что ИТ-отдел делал что-то «плохо», чтобы потом сделать хорошо. Это, безусловно, сказывалось на отношении руководства к подобным инициативам. Бывали ситуации, когда что-то переписывая, мы могли на несколько часов сломать часть основного процесса. А несколько часов простоя — это репутационные потери, отсутствие продаж, незаполненные самолёты и пустые отели. То есть очень серьезные деньги.

В предыдущие хлебные годы проблемы роста закрывались аппаратными мощностями. Принималось решение о закупке новых серверов, давался зарок, что это последняя закупка, а через полгода серверы приходилось закупать снова. Осенью 2015 года этот круг был разорван, и началась постепенная переработка программного обеспечения под новые цели. Основным ограничителем роста на тот момент являлась система хранения данных, и именно её было бы идеальным обновить за тихие зимние месяцы, когда на упавших продажах объемы данных растут минимальными темпами. Мы должны были заменить хранилище, добавить кэширующие подсистемы и снять ограничения на масштабирование — то есть фактически построить новую платформу.

История учит, что нельзя взять и создать с нуля новую платформу рядом со старой, а потом взять и заменить их в новогоднюю ночь.

Шок от проблем при запуске будет настолько велик, что после него можно и не восстановиться. Помимо этого, сложность системы потребует длительной разработки, и в это время все новые бизнес-требования придется реализовывать сразу в двух продуктах.

Изменения делались маленькими, очень осторожными шагами. Нельзя сказать, что ни один пользователь при этом не пострадал. Многие месяцы система работала медленнее обычного, с периодически выпадающим функционалом.

Такие долгие периоды стресса невозможны без четкого понимания руководством компании, зачем это надо и какие будут последствия. Без достаточного кредита доверия подобные реформы лучше даже не начинать, чтобы не остаться с полуразобранной системой в тот момент, когда руководство устанет и откажется платить за переделку уже существующего функционала.

Осторожный подход к переделке системы не позволил реализовать план за зиму, и замена хранилища в работающей системе проводилась как раз в момент летнего повышения нагрузки. Если проводить аналогии, это было эквивалентно замене двигателя в несущемся автомобиле. Однако внешние и внутренние пользователи системы этого не замечали — офисы работали, сайт работал, продажи шли, а в этот момент одна система хранения заменялась на другую.

Процесс обновления шел полтора года. Это было сложно и опасно.

Изменение модели

Когда в Штатах закончилась программа «Аполло», остались тысячи классных специалистов. И государство решило перевести их на производство бытовой продукции. Вот что написал по этому поводу один из ведущих разработчиков: «Когда они мне показали детскую игрушку, которую я должен производить, работа показалась мне пустячной. Затем я запросил температурный диапазон ее работы. Он оказался приемлемым. Потом я затребовал данные по

перегрузкам, ударам и вибрациям. Оказалось, что ребенок колотит игрушку так, что перегрузки впятеро больше, чем в космосе. Сделать это было трудно, но можно. Но когда они мне сообщили, сколько центов должна стоить такая игрушка – я понял, что сделать этого не смогу. В «Аполло» цену с нас не спрашивали».

Одним из способов выживания ИТ-отдела в условиях кризиса стал выход на внешний рынок и продажа своих продуктов и услуг в конкурентной среде. Однако анализ текущей рентабельности разработки показал ситуацию, подобную той, которую описал американский инженер.

Ситуация, когда у компании есть деньги на реализацию любого каприза менеджера, осталась в прошлом. Пришло время делать массовые проекты с минимальными затратами, что требует смены менталитета и методологии.

В «Библио-Глобусе» практикуется проектный подход к задачам. Бизнес формулирует свои требования, разработчик использует свое время для их решения. Разработчик получает зарплату за время, а не за количество закрытых задач. При этом разработчиков мало и стоят они дорого, поэтому компания заинтересована, чтобы они использовали свое время только на решение задач. Вспомогательные процессы, вроде рефакторинга, нормализации кода и доделок по инициативе самого разработчика, получают меньше внимания. На длительном забеге это приводит к потере качества кода и дублированию функционала.

При ведении одного, пусть и огромного, проекта стабильной группой разработчиков эти проблемы решаемы. Но в момент выхода на рынок, при росте количества проектов и внедрений, ситуация перестаёт быть управляемой, и себестоимость поддержки решений растёт экспоненциально.

«Расходы «Сбербанка России» на ИТ в 2014 году составили около 65 млрд руб., в 2015 году они выросли на 25–30%», – рассказал РБК «Расходы «Сбербанка России» на ИТ в 2014 году составили около 65 млрд руб., в 2015 году они выросли на 25–30%», – рассказал РБК первый зампред Сбербанка Лев Хасис. Это больше всех расходов компании «Яндекс» за аналогичные периоды. «Сбербанк – это крупнейшая в стране ИТ-компания. Мы занимаем 10% российского рынка ИТ, из 450 тыс. программистов 22 тыс. работают у нас», – отметил Греф¹.

¹ <http://www.rbc.ru/finances/15/01/2016/5698ce9d9a794791cf2c1748>

При этом остаётся задача обеспечения окупаемости. В «Библио-Глобусе» всегда отрицательно относились к раздуванию штата, и эту политику не планируется менять. Поддержка массового рынка с множеством внедрений за счёт увеличения персонала — кривая дорога, которая снова приведёт к росту себестоимости и потере управляемости. Приведенные во врезке слова Германа Грефа о 22 тысячах программистов в структурах «Сбербанка», возможно, означают низкую совокупную эффективность ИТ-департамента из-за сложности управления такой структурой плюс существенные даже по меркам «Сбербанка» затраты на персонал. И это не тот путь, которым идет наша компания.

В этом контексте изменение процессов и продуктов компании должно проходить своим, особенным путём. Мы не можем начать разрабатывать новые продукты с нуля по классическим канонам — на это нет ресурсов и времени. С другой стороны, компания вложила сотни человеко-лет в разработку платформы под свой туристический продукт, вместе с самим продуктом. Многообразие процессов внутри получившегося продукта таково, что ими можно покрыть несколько отраслей — так, в конечном счёте, и происходит. Параллельно, за время своего развития платформа научилась масштабироваться в обе стороны — как до размеров отрасли, так и до установки на миникомпьютерах.

Мы используем эти возможности, чтобы построить на той же платформе компонентную базу, из которой будут собираться новые продукты. Компонентом при этом становится не готовый функциональный модуль, как в других промышленных системах, а оформленный бизнес-процесс. Функциональный модуль нельзя приспособить в другой индустрии. Бизнес-процессы между индустриями могут совпадать на 90%, отличаясь только терминологией. Заложив в платформу возможность самостоятельно понимать связи между компонентами, мы получим независимый от индустрии конструктор, которому для промышленного запуска будет достаточно настройки не программистом.

При создании продукта под нового заказчика модульность на таком уровне абстракции требует участия программиста только при автоматизации новых, до сих пор не встречавшихся процессов. И с каждым случаем степень повторно использованных компонентов будет только увеличиваться.

Уже существующие процессы будут собираться вместе подобно паззлу, их настройкой и стыковой между собой сможет заниматься сотрудник более низкой квалификации. К примеру, автоматизация автозаправки включает в себя процессы:

- оплаты (кассы, кредитные карты, терминалы)
- контроля доступа (шлагбаумы)
- учета топлива (приход/расход)
- обеспечения безопасности (камеры наблюдения, кодовые замки)
- и т. п.

Разбирая кодовую базу туристического проекта по процессам, мы уже сейчас можем набрать компонентов на несколько десятков бизнесов самого разного масштаба.

Путь, по которому мы движемся, позволяет малой кровью решить несколько больших задач:

- исключить разрыв в текущей разработке. Туристический продукт остается на плаву, постепенно превращаясь из монолитного в компонентный;
- плавно формировать кодовую базу для построения новых продуктов и эффективно ее повторно использовать;
- снизить затраты на внедрение — как временные, так и на стоимость персонала.

Стратегически, двигаясь этим путем, мы создаем для бизнеса инструмент быстрой автоматизации при минимальных затратах. Но смена идеологии продукта — задача повышенного уровня сложности. Для её осуществления нам придётся реализовать следующие принципы:

1. Максимальная необслуживаемость. ПО не должно требовать настройки большей, чем может оказать неспециалист. Например, мы автоматизируем отель из 6 комнат в курортной зоне Сыктывкара. Там никто не будет нанимать системного администратора, ПО должно запускаться на том, что они назовут сервером. Но сейчас оно стоит на сотне серверов и требует полчеловека для обслуживания. Если добавить еще несколько сот серверов, то потребуется уже два человека — просто в силу объема. Поэтому система должна делиться на самостоятельные необслуживаемые (автоматически обслуживаемые) части. И это касается не только аппаратной, но и программной части системы. Программное обеспечение не должно требовать избыточных

ресурсов на обслуживание, оно не должно создавать ситуации, когда ошибка разработчика останавливает важный бизнес-процесс и проблему нельзя решить иначе, как срочно выдергивая этого разработчика на исправление ошибки. Речь идет именно о решении внезапно возникшей проблемы, а не плановом исправлении ошибок.

2. Круглосуточная доступность. ПО не должно требовать перерыва для своего обслуживания — в достаточно больших предприятиях работа идет круглосуточно, и остановка на регламентное обслуживание вызывает потерю денег и репутации. Если же ваша система связана с материальными средствами, то регламентное обслуживание может просто привести к коллапсу. Например, на одном мясокомбинате обновление системы регулярно, на несколько часов, приводит к остановке аппарата обработки собранной крови, и у них возникают вполне конкретные проблемы, куда девать несколько тонн этой крови — ведь на улицу не выльешь. Благо, если возможность остановиться есть, но рассчитывать на нее нельзя.

3. Полнота реализованного функционала. Чем больше сделано правильно, целостно, тем меньше дальнейших запросов на переделку будет получено и меньше времени на это потрачено. Речь не идет о слепом копировании процессов в отрасли или компании в новую систему. Для того чтобы все сделать правильно, придется сначала детально изучить и осмыслить всю схему. Только тогда станет понятно, какие шаги избыточны, а что необходимо. Когда в «Библио-Глобусе» приступали к стыковке с авиакомпаниями, подобный анализ не был проведен, и первые реализации не предполагали, что что-то может измениться. По мере увеличения числа стыковок появлялось все больше нюансов, не вписывающихся в спецификации и требующие заплаток и костылей. В результате вся «экосистема» стыковок несколько раз полностью переписывалась, чтобы получить достаточную гибкость для безболезненного подключения новых систем и чистки от мусора, который никем не используется.

4. Уникальный, трудноповторимый функционал. Методика, которая привела компанию на первое место в туризме, может быть распространена и на другие области. Более того, дополнительный эффект достигается за счет синергии наработок в разных отраслях — с этим вообще сложно бороться тем, кто специализируется только в одной области, а «Библио-Глобус» успел

реализовать проекты много где, пока проводил экспансию на соседние рынки. Сейчас эти решения интегрированы в общую систему, и вынесение независимых, но стыкуемых процессов — это одна из задач, которую надо решить.

5. Эластичность. Продукт должен уметь масштабироваться до размеров отрасли и обратно. Эту необходимость компания прочувствовала за лето-осень 2015 года, когда число вылетающих туристов варьировалось от 20 тысяч до 200 человек в сутки. Эластичность нужна на растущих проектах, а в непредсказуемых условиях она просто необходима. Например, реализация проекта единой турпутевки, обсуждаемая правительством, потребует резкого прироста вычислительных мощностей, и редко какая готовая система сможет вырасти на всю страну без коренной переделки.

6. Открытый API. За годы деятельности «Библио-Глобус» использовал множество чужих API и создавал свои. Сейчас у компании уже есть понимание того, что и как должно быть при передаче туристической информации, и мы готовы этим знанием делиться. «Библио-Глобус» намерен сформировать и продвигать открытый API для своих продуктов, в первую очередь, туристических, стараясь сделать их базовыми в стране, а затем значимыми в мире. Теоретики, формирующие стандарты, не выдерживают борьбы со временем и проигрывают практикам. К тому же история показывает, что выживают только те стандарты и технологии, которые поддерживаются большими компаниями.

При реализации этих принципов мы приходим к модульной системе управления ресурсами. Модульной на всех уровнях — от хранения данных до бизнес-процессов и вспомогательных подсистем. Обеспечив в ней связность по данным, мы получаем систему управления ресурсами произвольного масштаба, которая может настраиваться и обслуживаться менеджерами, а не техническими специалистами.

Предоставляя людям инструменты и возможность их настройки, мы получим «экосистему», самостоятельно поддерживающую дальнейшее развитие продуктов. Ни одна компания (кроме специализированных) не может поддерживать большой объем внедрений, это должны делать сами пользователи, обмениваясь конфигурациями и знаниями.

Много труда требует подготовка системы хранения, которая с минимальными настройками должна «пережевывать» данные самых разных объемов. Необслуживаемость системы зависит от объема данных, поэтому одна из задач – резкое уменьшение количества хранимых данных, например, за счет оптимизации их расчета из каких-то других данных или же кластеризацией больших хранилищ на много необслуживаемых маленьких. Когда данных мало – все хорошо, быстро, и система может работать годами без обслуживания. Если же все работает на пределе – постоянно требуется вмешательство или контроль человека.

Но не меньшее внимание требуют интерфейсы работы с пользователями. Сила и сложность системы должны быть скрыты за возможностью быстро настроить требуемую конфигурацию.

«Библио-Глобус» исторически силен платформой – серверной стороной технологического стека. Акцент на фронтенде не ставился никогда, и это создало определенную репутацию у пользователей. Намного хуже было бы при обратной ситуации. Технологическая пирамида не может стоять на голове, и те, кто делает фронтенд, не выстраивая под ним серьезный бэкенд – обречены упереться в потолок при достижении пороговой нагрузки.

Поэтому к созданию гибкого и настраиваемого фронтенда мы подошли довольно поздно. Для работы с нашими данными был создан и развивается механизм настройки ввода и отображения данных, пригодный для использования обычным пользователем. Сейчас эта технология уже реализована в интернет-версии книжного магазина «Библио-Глобус» и проходит тестирование перед промышленным запуском. Мы постепенно переводим на неё внутренние административные интерфейсы и вспомогательные подсистемы.

Итогом работ по перестройке системы «Библио-Глобуса» становится платформа, способная растягиваться до отраслевых масштабов, и позволяющая осуществлять все процессы, не предъявляя высоких требований к ИТ-квалификации пользователя для достижения максимального охвата.

Запрос на подобные большие системы остается высоким в отраслях и компаниях, работающих с чувствительной информацией. Никакие больницы, электростанции и пенсионные фонды никогда не будут хранить свои данные в

чужом облаке, несмотря на удобства и объемы. Им нужна подобная система, установленная локально, и желательно, собранная из дешевых компонентов.

Сейчас наше развитие идет в двух направлениях: информационные системы и программно-аппаратные комплексы — квадрокоптеры и лазерная оптика. Аппаратные решения еще не вышли на серийное производство, чтобы о них можно было говорить как о состоявшемся производстве. А если говорить о продаже информационных систем, то оказывается, что при необходимости можно собрать достаточно заказов на всю команду «Библио-Глобуса» и еще два таких коллектива. Репутации компании достаточно, чтобы мы могли выбирать, чем заниматься. Мы способны прокормить свой отдел разработкой ПО, это стало понятно в тот момент, когда мы анонсировали свою открытость к внешним проектам.

Прошедшие два года уже позволяют нам делать выводы о своих сильных и слабых местах. Уже можно сформулировать, чем мы должны и будем заниматься, а чего стоит избегать и с чем не связываться вообще:

1. Мы не должны брать деньги в долг. Есть множество примеров, как долги приводят компании к катастрофическим последствиям. Заёмные средства приводят к росту ценой потери независимости. Как бы медленно мы ни развивались, это будет происходить на собственные деньги.

2. Мы не должны сами заниматься продажами. Для этого в штате должны быть специальные люди, чья работа состоит в создании ажиотажа. Вместо этого мы нашли желающих стать нашими дилерами и продавать наши продукты. Подход сработал — на наш лазерный прибор в цепочке продаж выстроилось четыре человека, каждый со своей наценкой, конечная стоимость продукта выросла почти в два раза, но эта цепочка нашла нам заказчика.

3. Мы будем заниматься стандартами. Начиная с туроператора, там, где мы получаем серьезную долю рынка, описывать стандарты, интерфейсы, выставлять их на общее обозрение, понижая сложность и уменьшая стоимость входа, и забирать в свою зону влияния огромное число тех, кто с нами хочет работать. Таким образом, действовать на опережение. Созданию полноценной документации исторически уделялось мало внимания, но эта

проблема относится к числу решаемых.

Придется описать стандарты и протоколы подключения к нашим системам и выгрузки к нам данных (например, в случае туроператора, цен), чтобы поставщики услуг могли самостоятельно создавать стыковки со своим ПО и автоматизировать работу с нами. У нас уже есть API для выгрузки данных, а API для приема данных еще нет, но потенциал в этом огромный.

4. Мы будем нести людям добро. Разрабатываемые стандарты и протоколы должны быть открытыми, а сборки компонентов для бизнеса — бесплатными. Мировой опыт показывает, что такие бизнес-модели живучи и не причиняют ущерба компании, а развитие «экосистемы» продукта осуществляется сформировавшимся вокруг него сообществом. Закрытые стандарты не позволяют получить поддержку от сильных игроков на рынке.

Наши сильные стороны — наработанная база решений и отлаженная методология, описанная во второй главе. Чтобы не становиться одной из многих компаний, автоматизирующих что угодно, мы поднимаем планку и строим автоматизацию следующего уровня.

В идеальном мире на любом уровне управленческой цепочки сотрудник обладает всей полнотой информации. В реальном — чем выше, тем дальше надо готовить цифры и меньше они совпадают с реальностью. Поэтому мы считаем востребованными и будем заниматься двумя новыми направлениями — аналитикой по руководству проектами и экономикой проектов.

И наша страна, и весь мир сейчас выходят на очень интересный этап развития. Если под автоматизацией в России на большинстве предприятий понимается избавление от ручного труда, то в «Библио-Глобусе» эту стадию давно прошли, и мы планируем развивать и продавать её следующий этап, который определяется как «автоматизация — это эффективность». Эффективность не отдельно взятых автоматизированных процессов, а организации в целом.

Попытки предоставлять данные в реальном времени для принятия управленческих решений проводятся с начала эры вычислительных машин с переменным успехом. Можно вспомнить проект Cybersyn Стаффорда Бира или ОГАС (см. труды академика Глушкова В.М.) В настоящее время технологии, в отличие от людей, уже созрели для качественных изменений процесса управления.

Через несколько лет технологии позволят проводить аналитику бизнес-процессов в реальном времени, и владельцы бизнеса смогут принимать решения на основе полной информации. И не будет необходимости в посредниках, которые формируют отчеты и собирают данные для принятия управленческих решений.

Возьмем показательный, но правдивый пример. Есть горнолыжный курорт. У подножия горы сидит кассир и продает скипасы (пропуска на подъемник), причём часть он продает мимо кассы. Чтобы это пресечь, к нему надо приставить охранника и платить тому зарплату. Охранник оказывается еще более коррумпированный, он берет кассира в долю, и они уже вдвоем продают скипасы туристам. Для присмотра за ними нужно брать контролера, что ещё более раздувает штат. Помимо того, у кассира есть начальник, главный кассир, который тоже получает зарплату. Решение в стиле «автоматизация есть избавление от ручного труда» — поставить вместо них терминал, пусть туристы покупают скипасы сами. Правда, сделать такой терминал курорт сам не в состоянии, для этого нужно иметь определенную инфраструктуру и наработки.

Подобная система не только коррумпирована, она еще и неэффективна. Информация обязана быть доступна оперативно, а не из месячных финансовых отчетов, и это критически важно для индустрий, где за полчаса можно поставить рынок с ног на голову. Определением эффективности становится возможность получать в реальном времени данные о движении денег и товаров не только за прошедший период, но и в экстраполяции.

Продолжая историю с курортом, скажем, что дальнейшая автоматизация продажи скипасов должна давать владельцу оперативные данные продаж за сегодня, среднее время на склоне, текущую загрузку подъемников, иметь возможность регулировать цену в реальном времени в зависимости от всех этих параметров для поддержания заданных показателей доходности/нагрузки/комфорта.

Если брать промышленное производство, это означает видеть у себя на мониторе в каждый момент времени с детализацией до любой величины, как изготавливается продукция, какую операцию делает каждый человек в данный момент времени. Какой процент общего изделия он сделал, сколько это изделие сейчас стоит, и сколько оно будет стоить на выходе. На обычных производствах такая информация отстает по времени весьма существенно, а

о прогнозировании вообще сложно говорить.

Руководителям разного уровня должна быть доступна информация с разным уровнем детализации. Число моторов на складе может быть важно начальнику сборочного производства, а руководитель компании должен видеть, что у него сейчас заложено три вертолета на стапелях, два практически готовы к сдаче, и еще два сделаны на 25 процентов. Он должен знать, что на производстве каждый человек загружен, этот перерабатывает, тот недорабатывает, что каждая операция стоит столько, и что в такой-то момент времени будет готово пять вертолетов по рассчитанной цене. То есть по себестоимости плюс прибавленная работа. Но прибавленная работа вычисляется не классическим умножением на два, а по зафиксированным человеко-часам.

Эту методологию как часть консалтинга нельзя продать без рабочих примеров. Поэтому первое внедрение придется на сам «Библио-Глобус», его туристический продукт.

Помимо новых разработок, мы продолжаем фокусировать внимание на интеграции со сторонними системами. Несмотря на все амбиции, компания сейчас не в тех условиях, чтобы развернуть свои разработки на всю страну. Поэтому нам предстоит период стыковок еще со множеством разных систем – банки, платежные системы, билетные системы, поставщики и потребители ресурсов. Имея множество самых разных входов и выходов к нашей системе, мы можем легче замыкать между собой совершенно разных потребителей услуг.

Сейчас каждая качественно новая стыковка разрабатывается с нуля. Идеальным решением было бы найти общие точки в процессе, чтобы стыковка стала одним из компонентов системы – параметризуемым и настраиваемым. Речь идет как об интеграции с программными, так и с аппаратными системами. Микроконтроллеры, подсистемы IoT (интернета вещей) и подобные им являются крайне перспективными, и учиться работать с ними надо уже сейчас. Для их обкатки у нас есть производство квадрокоптеров, которое стыкуется с нашим же оптическим комплексом, и для которого логичным развитием будет создание стандартов интеграции со службами мониторинга, доставки, спасения и подобными.

Как часть интеграции с внешним миром собственные ресурсы компании

должны стать открытыми. Мы должны предоставить потребителям API для доступа к своим ресурсам, чтобы переложить на них часть работы. В тех отраслях, где компания доказала своё преимущество (в первую очередь, туризм), мы уже можем формулировать правила. И API — это правила работы с нами. Если агентству что-то от нас нужно — у нас есть документация, пусть подсоединяются и работают, всё в открытом доступе. Мы тратим меньше денег, а агентства соревнуются за качество сервиса и функционал, которые они могут построить на наших данных.

Для этого, разумеется, нужно много времени и внимания уделить как самому API, так и документации для него. Ранее этой задаче не ставились высокие приоритеты, мы, скорее, реагировали на запросы рынка о том или ином функционале. Сейчас есть возможность создать отраслевые стандарты для нормальной работы всех участников. Наш собственный опыт стыковок говорит, что единый стандарт является хорошим драйвером роста конкуренции в отрасли, что для самой отрасли весьма полезно.

Более того, мы вынуждены заниматься индивидуализацией и персонализацией при работе с клиентами. Рынок уверенно переходит от модели B2B к B2C. Когда «Библио-Глобус» работал только с профессионалами, ему было простительно иметь «несовременный» пользовательский интерфейс, поскольку тогда выигрывала компания с самой сильной и функциональной платформой, а не с самым удобным сайтом. Сейчас количество заходов на сайт с портативных устройств (хороший критерий частного клиента) достигло некой критической величины, после которой игнорировать их уже непростительно. На Youtube есть ролики «как бронировать на сайте Библио-Глобуса», и неспроста. Эта сложность должна уйти, и поэтому мы начинаем заботиться о наших прямых покупателях, одновременно избавляясь от потерь на комиссиях посредникам.

В туристическом направлении мы целенаправленно выстраивали вертикальную линейку продуктов. Недавно мы взяли во фрахт самолеты и сейчас контролируем авиационные процессы. Мы уже не только туроператор, но и авиакомпания, пусть без названия, но по факту это так. Нам нужно выходить на самостоятельные розничные продажи для замыкания вертикали: авиакомпания — отель — туроператор — розница.

И, формируя наши новые продукты, мы должны научиться работать с масштабированием. Масштаб системы должен соответствовать продукту на

любом этапе его жизненного цикла, и наша задача научиться делать системы, которые растут вместе с продуктом. Идеально, если клиенту ставится маленькая система, а когда его бизнес вырастает до масштабов страны — ему нет необходимости менять систему, она продолжает справляться с растущими объемами данных и сложностью процессов. То есть продолжает подходить под задачи, и на каждом этапе развития предприятия клиент может без усилий подключать необходимые бизнес-процессы.

Имея такие планы, ИТ-команда неформально была вынуждена сменить структуру, став небольшим заводом по производству программных продуктов. Новое функциональное деление можно описать так:

1. Разработчики платформы. Они определяют базовую функциональность системы. Их задача — формировать компоненты на основе повторно используемого функционала и обеспечивать их работу на разных сборках системы. Планировалось, что для начала компоненты будут выделяться из туристической системы «Библио-Глобуса», параллельно обрастая возможностями кастомизации. Но оказалось, что туристический монолит просто так не расколоть, и пока компоненты строятся на основе новых разработок.

2. Разработчики бизнес-приложений. Это люди, собирающие решения под поставленную задачу — как из уже имеющихся компонентов, так и дописывая необходимое ПО. Сюда входят и фронтенд-разработчики, и дизайнеры, поскольку каждое решение должно нести свою уникальность. Для поддержки универсальных решений мы пытаемся вести клиентскую часть на единых принципах и стандартах, вытравливая устаревшие решения по мере обновления приложений и веб-страниц.

3. Методологи. Те, кто общается с внешними заказчиками и формулирует требования к системам. Им достаточно понимать философию построения систем на нашей платформе для корректной формулировки задач разработчикам.

4. Студенты и стажеры. Мы пока не доросли до открытия собственной академии, но нашли применение студентам, которые после дополнительной стажировки помогают остальному коллективу в решении множества небольших, постоянно возникающих задач. Не имея специализации, они

постепенно изучают разные части системы, приобретая по ходу работы общее понимание её сложности.

При наличии заказчика полученная структура может действовать вполне самостоятельно, обеспечивая результат без необходимости постоянного контроля и постановки детализированных задач руководством. Руководство может уделять больше внимания другим задачам и направлениям.

РЫНОК

Поиск внешних заказчиков привёл к достаточному количеству предложений, чтобы понять, что у команды уже есть репутация и вне туристического рынка. 31 марта 2016 года первый внешний продукт «Библио-Глобуса» — электронная торговая площадка — был презентован на Российско-Сингапурской конференции по сотрудничеству. Таким образом, «Библио-Глобус» стал компанией - разработчиком, начав с международного проекта.

В партнерстве с «Опорой России» мы запустили поисковую площадку Go2RUS, ориентированную на внутренний туризм.

Затем последовало ещё несколько проектов отраслевого масштаба, включая государственные. Государству тоже нужны реально работающие вещи, а на практике реализовать что-то работающее, а не только заявленное, могут не так много команд.

Тезис «сначала вы работаете на репутацию, потом репутация работает на вас» подтвердился в полной мере, когда сам рынок начал приходить и спрашивать про ещё незавершенное ПО для автоматизации гостиниц. Люди каким-то образом узнавали, что ведётся разработка, и обращались к руководству с запросами на установку системы в их гостиницах. При этом никто, включая руководство, еще не видел продукта и не знал о его качествах и возможностях. С этим продуктом не надо пробиваться на рынок, он уже в рынке, пусть даже не написанный.

А в конце 2016 года, как уже говорилось во второй главе, компания взяла в управление гостиницу «Азимут Отель Сочи» (теперь «Сочи Парк Отель») на 2880 номеров, запустила в неё команду своих ИТ-специалистов, и начала автоматизировать в стиле «Библио-Глобуса». Для обеспечения эффективности

управления нам потребуется пересмотреть буквально каждый процесс, начиная с закупки картошки для кухни (цена может меняться от 5 до 200 руб/кг, на объемах гостиницы – колоссальные деньги) и заканчивая системами безопасности. Для нас проект такого масштаба – источник целого ряда бизнес-решений, которые потом успешно можно поставлять другим клиентам. Терминалы по продаже услуг, платные парковки, гостиничное ПО, автоматизация деятельности ресторанов и прочее – нам банально не хватает людей, чтобы покрыть всё сразу.

Первая проблема неавтоматизированной системы – не все данные есть в виде, пригодном к обработке. Чтобы приступить к анализу, сначала придётся оцифровать договоры, счета, заказы и номенклатуру. На базе этой номенклатуры придётся завести склад для контроля и учета остатков. Как уже описывалось во второй главе, у нас появляется прозрачность процессов и денег, у партнеров снижается возможность идти в обход системы. А в дальнейшем мы запускаем аукционную модель для поставщиков в гостиницу и попытаемся уйти от сговорных отношений между закупщиком и поставщиком.

Заключительное слово

Уровень автоматизации, существующий в компании к началу кризиса, позволил ей пережить ударные нагрузки последних двух лет без существенного сокращения штата. Тотальная автоматизация в совокупности со скоростью принятия решений руководством позволила крайне быстро реализовывать изменения стратегии для спасения ситуации.

При этом бремя экстренных переделок и доработок легло на ИТ-отдел, который выдержал проверку стрессом не развалившись и не переругавшись. Задач предсказуемо стало больше, сроки жестче, регулярность поступления новых вводных стала обыденностью.

Но компания выстояла, и стала крепче. Мы уже не боимся новых кризисов. За это время мы смогли омолодить систему и запустить новые продукты как в туризме, так и в смежных областях. Когда в конце лета 2016 года снова открылась Турция, туристический рынок выдохнул, и снова стало можно работать спокойно. Мы смогли снова сфокусироваться на прерванных планах.

Упомянутая уже B2C модель — это не только удобный клиентский интерфейс. Имея массу подключенных сервисов от сторонних организаций, мы запускаем возможность компоновать тур из наборных услуг, формируя индивидуальное предложение для клиента. При этом мы остаемся туроператором массового туризма, но уже с человеческим лицом. Вообще, тема персонализации в индустрии еще далеко не проработана, еще ждут своих героев такие области, как поиск информации о клиенте, чтобы на ходу сформировать максимально подходящее ему предложение; анализ неформализованных запросов; «умная» агрегация и выдача результатов. В какой-то момент мы обязательно до этого доберемся.