

16 июля 2021

# Ошибки техподдержки, и как с ними бороться

Чтобы поддержать рост компании, одна из основных ее систем — ERP — должна постоянно меняться. Однако зачастую эти изменения заключаются в постоянном «допиливании» отчетов и процессов по требованию функциональных заказчиков без оглядки на методологию, современные возможности систем или даже на стратегию бизнеса. Спустя годы ERP превращается в монструозный и дорогой в обслуживании самописный инструмент. Наш эксперт Константин Смирнов рассказывает, как не прийти к этой точке, и какие есть ключевые ошибки в поддержке и развитии системы.

Многие компании воспринимают техподдержку как сервисную функцию, заточенную на исправление ошибок. Это те люди, которые помогают сотрудникам использовать ИТ-технологии компании, решать возникающие технические проблемы, поддерживать производительность систем.

Будем честны, поддержка не всегда погружается в стратегию бизнеса, а руководители не очень понимают, как айтишники им могут с ней помочь.

Говоря о ERP, мы всегда подразумеваем операционную эффективность, реализацию целей компании. Это та система, которая требует особого подхода при внесении изменений — она отражает суть бизнеса и его процессов, влияет на работу других решений.



На практике развитие часто скатывается в поиски простых и быстрых шагов: «допилить» так, как требует бизнес, и закрыть заявку. Постепенно систему становится сложно и дорого обслуживать, а от слов «развитие», «интеграции», «обновление» всем становится не по себе.

Работать с ERP «напильником» — в некоторой степени кощунство, которое в будущем может вылиться в большие проблемы. Я остановлюсь на наиболее неочевидных и сложных ошибках и расскажу, как их исправить.

## **Несоблюдение методологического качества изменений**

Взаимодействие между ИТ-специалистами и подразделениями, которые вовлечены в создание продукта и генерацию прибыли, не всегда налажено. Порой доходит до того, что бизнес начинает самостоятельно внедрять инструменты. ИТ-отдел при этом остается в неведении.

Мы наблюдали случаи, когда рассинхронизация приводила к серьезным проблемам. В одной производственной компании маркетологи приобрели сервис для обработки данных клиентов. Спустя время поставщик ПО обанкротился и перестал поддерживать продукт. Маркетологи, потерявшие доступ к критически важной информации, обратились к коллегам из ИТ, но они уже ничем помочь не смогли. Бывают обратные ситуации, когда ИТ-специалисты ставят эксперименты над компанией.

Крупное предприятие, один из лидеров в своей области, в штате трудятся более двух тыс. человек. ИТ-отдел долгие годы кастомизировал ERP-систему, наращивал и изменял её функциональность. В какой-то момент ее

поддерживали 200 специалистов подрядных организаций. При этом финансовая служба вела налоговый учет в Excel, потому что система не отвечала требованиям бизнес-пользователей.

Когда нас пригласили провести обследование для запуска новой ERP, мы увидели огромную самописную систему, разобраться в которой было крайне сложно. В промышленную эксплуатацию на 900 пользователей была развернута даже не полноценная ERP-система, а ее бета-версия. Оказалось, что ИТ-специалисты компании тестировали, им было интересно поработать именно так.

Необходимость в изменениях систем и процессов будет всегда, это реалии рынка. Специалисты, которые инициируют преобразования, часто не знают, как их реализовать технологически, а люди с экспертизой в ИТ не всегда в курсе рыночных трендов. Стыковка между бизнесом, данными, программным и аппаратным обеспечением и есть методологически правильная организация изменений.

Подобный подход к управлению ИТ-архитектурой называется TOGAF (The Open Group Architecture Framework). Он рассматривает предприятие как комплексную систему, включающую четыре блока: бизнес-процессы, данные, информационные системы, инфраструктуру.

Любые изменения в одном из блоков должны быть согласованы и продуманы на всех остальных уровнях: что это даст бизнесу? Какие данные потребуются и возникнут ли новые? Достаточно ли аппаратных или облачных ресурсов для того, чтобы это все работало? Это помогает предусмотреть риски и не допустить изменений ради изменений.

Решение: привлекайте ИТ-архитектора или архитектурный комитет при планировании работ на период (например, квартал). Это может быть как внешняя экспертиза, так и внутренние специалисты, которые понимают, как устроена компания с точки зрения бизнес-процессов, данных, систем и инфраструктуры.

## Избыточная кастомизация

Вендоры разрабатывают свое ПО на основе лучших отраслевых практик и с привлечением экспертной поддержки (например, «1С» сотрудничает с компаниями из «большой четверки»).

Есть вендоры, которые тратят человеко-годы на создание функциональности, и есть ИТ-специалисты компании, которые считают, что готовые решения не подходят и нужно автоматизировать по-своему.

Часто это объясняется спецификой бизнеса и ноу-хау, которое невозможно переложить на типовую функциональность. Так снова получается сложная система, работа которой зависит от конкретных людей, что несет массу рисков для бизнеса.

У компаний нет ресурсов для проектирования и разработки на уровне вендора, а у того, в свою очередь, нет глубоких знаний конкретного бизнеса, но есть готовое решение, которое основано на опыте флагманских компаний и соответствует требованиям современного рынка.

Используя типовые функции, компания сокращает затраты на поддержку и получает гарантию от производителя ПО, что система будет стабильной и

производительной. Большое количество доработок, во-первых, повышает бюджет внедрения, во-вторых, влияет на совокупную стоимость владения (ТСО).

Решение: изучайте типовую функциональность и ищите баланс между бизнес-требованиями и существующими возможностями систем.

## **Отсутствие контроля за качеством кода**

Бизнесу сложно следить за качеством кода. Впрочем, ему не особо важно, сколько строк программист написал, если это работает в рамках требований к производительности. Низкое качество обычно представляет опасность именно для нее.

Условный пример: в супермаркете важно, чтобы чек на кассе печатался меньше, чем за 1 секунду. Если на рабочем месте кассира система будет работать медленнее, возникнет невероятная очередь, поэтому решение должно выдерживать обозначенную планку.

Для контроля есть большое количество инструментов: например, проведение нагрузочного тестирования и замеры производительности. Можно имитировать нагрузку в тысячи пользователей, сотни тысяч транзакций в момент, но сначала нужно определить требования к производительности, и здесь бизнес может обозначить и контролировать свои метрики.

Низкое качество кода — чаще всего проблема организационного характера. Когда двух программистов перегружают задачами и при этом грозят увольнением, результат будет соответствующим. Решать проблему нужно

путем создания системы управления разработкой, обучением и аттестацией специалистов.

Решение: создание системы управления разработкой. Она включает процедуры контроля качества, согласование архитектуры, тестирование и применение доработок.

Если бизнес хочет принимать непосредственное участие в цифровой трансформации, то для этого есть перспективное направление low-code. Такие платформы позволяют обычным бизнес-пользователям настраивать и модифицировать ПО без глубоких знаний программирования.

Вместо традиционной разработки low-code решения используют визуальные методы: сотрудники могут вносить изменения в специальном интерфейсе с помощью «перетаскивания» компонентов и других интуитивно понятных инструментов. Это ускоряет темпы цифровизации и в целом сокращает затраты на ИТ-персонал.

## **Зависимость от подрядчика и конкретных специалистов**

Любые изменения в системе должны быть отслеживаемыми. В вопросах технической документации лучше всего руководствоваться принципом разумной достаточности: вести реестры модификаций, поддерживать актуальную архитектуру процессов и систем хотя бы на верхнем уровне. Если подрядчик оставит вас, вам нужно быстро разобраться, как все устроено и найти замену.

К примеру, крупная компания лишилась разработчика, который поддерживал зарплатную систему. Без документации бизнес не понимал, что делать, а время сильно поджимало – нужно было платить сотрудникам зарплаты. Поэтому для независимой и стабильной работы важно обеспечить резервирование, своевременное обновление систем и необходимый уровень документации. Все это базируется на трех принципах: разумная достаточность, простая и регулярная актуализация, а также декомпозиция (описать все и сразу невозможно, разбивайте на блоки).

Решение: добавьте регламенты в процедуру обновления систем, ведите документацию с разумным уровнем детализации.

## Отсутствие SLA

SLA – это согласованные правила игры. Без этого документа можно столкнуться с комплексом проблем: низким качеством кода, игнорированием методологии, неоправданными затратами на доработки. Когда взаимодействие не регламентировано, стороны так или иначе будут перетягивать одеяло.

SLA регулирует сроки исполнения запросов и устанавливает между бизнесом и командой поддержки единое понимание критичных процессов. Возникает ответственность, вместе с ней и уверенность в стабильности систем. К SLA можно привязывать KPI, тем самым повышая уровень сервиса.

Решение: пропишите в договоре согласованный SLA и регламент взаимодействия, обговорите с сотрудниками KPI, привязанные к мотивации.

## Непрогнозируемые затраты на поддержку и развитие

Планируя изменения в ИТ-архитектуре предприятия, необходимо постоянно сверяться со стратегией. К нам обратились две компании с абсолютными противоположными запросами, но одинаковой проблемой — устаревшей ERP.

В первом случае у бизнеса есть желание стать лидером в своей отрасли за счет высокого уровня автоматизации. Компания сформулировала свой образ будущего, точно знает куда двигаться и ищет ИТ-партнера, который найдет оптимальные и быстрые способы прийти к этим целям.

Во второй ситуации ИТ-директор пришел к пониманию, что компания медленно уходит с рынка. У бизнеса больше нет запроса на развитие, поэтому он принял решение свести поддержку к минимуму — дорабатывать систему, только если возникнут новые требования законодательства к формам документов.

У бизнеса и ИТ должен быть единый взгляд на будущее – как с точки зрения технологий, так и развития компании в целом. Для этого требуется постоянная сверка задач, совместное обсуждение и планирование работ. Рекомендуем наращивать внутреннюю экспертизу — нужны люди, которые хорошо знают бизнес и его стратегию, умеют «фильтровать» запросы и принимать сложные решения.

Решение: планирование объема работ на период (квартал) при участии бизнеса и ИТ. Сверка со стратегией компании.

## Скептическое отношение к внешней экспертизе

Иногда развитию мешает банальная зашоренность, особенно когда возникают новые, нестандартные рыночные условия. Нельзя забывать про человеческий фактор: специалисты устают, теряют интерес.

Любые кросс-отраслевые практики и знания могут дать интересные идеи для изменений на всех уровнях. Пример из реальной жизни: я приобрел кроссовки онлайн, а забрать мне их предложили в магазине зоотоваров. Заодно купил собаке игрушку. Все участники кросс-продажи оказались в выигрыше.

Узкий взгляд часто оправдывается специфичностью бизнеса, но окопная методика в наших реалиях может стать ограничительным фактором: без экспериментов не выжить, ты стагнируешь и уходишь с рынка.

Решение: привлечение внешней экспертизы с более широким опытом работы с разными заказчиками.

Служба поддержки как функция имеет большой потенциал. Она может быть стратегическим ИТ-партнером бизнеса, инициатором и куратором технологических преобразований, влиять на прибыль компании.

На самом деле для «перезапуска» поддержки многие вопросы уже решены — специалисты знают ИТ-системы компании, отношения с функциональными заказчиками установлены, а значит, наладить взаимодействие бизнеса и ИТ возможно.

Ваша техподдержка готова отказаться от «напильника» и взять на себя новую роль?

*Источник: Rusbase*

