

Ключевые вопросы выбора типового решения для создания ERP-системы предприятия

Н.Г. Лисин (ИТПП)

Рассматриваются ключевые вопросы выбора комплексного ERP-решения для автоматизации управления производственным предприятием. В качестве основных критериев выбора обозначены сроки поддержки ПО разработчиком и возможности настройки решения в соответствии с принятыми на предприятии бизнес-процессами. Предлагается выносить специфические функции, характерные для бизнес-процессов конкретного предприятия, на отдельный логический уровень, благодаря чему ядро системы может остаться неизменным и сохранить полноценную поддержку разработчика. Приведены практические примеры, реализованные на платформе 1С.

Ключевые слова: автоматизация управления производством, бизнес-процесс, ERP, типовое решение, программная доработка, поддержка программного продукта.

Руководство все большего числа предприятий в последнее время принимает решение радикально обновить информационную систему организации с целью общего повышения эффективности деятельности, минимизации производственных издержек и, как следствие, увеличения прибыльности. Критериев выбора оказывается множество. А "муки выбора" поставщика решения — это типичная ситуация в начале проекта.

Определиться действительно бывает сложно. Каждый поставщик пытается показать себя с лучшей стороны, демонстрирует свой богатый опыт и достижения, умалчивая о проценте неуспешных проектов и о типовых трудностях, возникающих в ходе адаптации систем к требованиям предприятия.

Рассмотрим ряд важных моментов, которые обязательно необходимо учитывать в ходе выбора стратегии и тактики создания автоматизированных информационных систем на предприятии. Ниже не только зафиксированы возможные сложности, но и показаны критерии выбора типовых решений, использование которых позволяет обходить эти проблемы. Все проблемы рассматриваются на примере историй из реальной практики.

Привычное — враг разумного

Практический опыт многих проектов подсказывает, что существуют достаточно очевидные и разумные способы повышения прогнозируемости результата при выборе поставщика решения. Однако эти способы выходят за рамки сложившихся у большинства ИТ-руководителей стереотипов. Другими словами, для достижения адекватности подхода важно осознать, что "разумное" по рамкам гораздо шире, чем "привычное".

Рассмотрим практический пример. Как правило, поставщики предлагают наиболее тиражируемый продукт и свой опыт его внедрения в неких отраслях. Опыт решения аналогичных задач — это основной критерий, однако выбор происходит в плоскости только одного продукта.

Ограничивая свой выбор плоскостью единственного продукта, представитель предприятия должен понимать, что будет связан свойственными этому продукту ограничениями функциональности. И эти ограничения могут быть принципиальными и "пожизненными".

Дело в том, что в любом типовом решении заложена не вся предметная область и не все возможные бизнес-процессы, а только опыт, на основании которого разрабатывалось и далее развивалось типовое решение. Этот опыт по определению не безграничен, причем ключевые "узкие места", заложенные в решение при его начальной разработке, могут очень серьезно ограничить его дальнейшее развитие.

Причина данной проблемы в том, что разработчики не могут перекраивать продукт как угодно для достижения новых качественных свойств. Любая новая версия программы должна обеспечивать полную совместимость с теми данными, которые накопили клиенты в предыдущей версии. Иначе клиенты, купив один раз программный продукт, уже не смогут загружать новые версии с новыми функциональными возможностями. Фактически начальная разработка задает "туннель" ограничений, по которому движется решение в процессе своего развития. Этот туннель — структура данных. А заказчики, которые ведут поиск только в плоскости одного продукта, ограничивают себя в вариантах выбора.

Важно "подняться над плоскостью"

В космофизике есть интересная аналогия — описание двумерного "плоского" мира, который противо-поставляется реальному трехмерному. Жители "плоскости" перемещаются только в рамках двух координат, не подозревая о существовании третьего измерения. И если, например, на их пути возникает препятствие, они вынуждены его обходить, хотя гораздо проще было бы перепрыгнуть, но для этого нужно использовать третье измерение, которое им недоступно. Так же ведет себя и предприятие, ограничивающее свой выбор плоскостью одного типового решения.

Любое ограничение — это упущенные возможности. Удивительно, что лица, принимающие решения, добровольно ограничивают, сужают свои возможности и возможности сделать проект автоматизации результативным.

Привычные подходы — так ли они хороши?

Доработка типового решения под требования бизнес-процессов

Основное преимущество типовых решений 1С состоит в том, что эта система абсолютно открыта, содержит встроенный язык программирования, благо-

*Никто не растет сам по себе.
Всегда есть информационные технологии,
которые влияют на развитие предприятия.*

Журнал "Автоматизация в промышленности"

даря чему решения на платформе 1С обладают гибкостью. У предприятия всегда есть возможность купить типовое решение и путем доработок "заставить" его выполнять любые задачи специфичные для конкретного предприятия. Например, после осуществления необходимых доработок бухгалтерская программа получит функционал планирования производства, а складская программа сможет решать задачи страхования и управления недвижимостью.

Именно поэтому функциональные возможности типовых решений с точки зрения логики предметной области не становятся ограничением в их применении. Вся нужная функциональность дописывается на местах, неважно, реализуется ли в системе специфика заказчика или устраняется недостаточность общей функциональности типового решения. В итоге такой "тонкой доводки" предприятие получает именно тот продукт, который ему нужен. Казалось бы, ситуация складывается оптимальным образом, но...

Актуальным остается вопрос, насколько долго сохранится достигнутое по результатам внедрения качество такой заказной системы? Каковы будут перспективы ее развития и гибкость? Также нельзя забывать о финансовой составляющей проекта и стоимости программных доработок на начальном этапе и в ходе пользования системой.

А самое главное — если доработанная программа будет отличаться от типовой, можно ли будет загружать в дальнейшем новые версии программы от поставщика, актуализировать изменения в регламентированном учете? Ведь программисты предприятия (или компании-внедренца) сделали в некотором наборе модулей системы свои доработки, а в то же самое время разработчики программы постоянно в этих же самых модулях что-то меняют, исправляют и дополняют, ведь система непрерывно развивается. Да и правила регламентированного учета время от времени меняются, что требует отражения в программе.

Будут ли совместимыми проектные доработки? Можно ли будет загружать обновления от поставщика, не потеряв того, что сделали на предприятии "под себя"? Какова будет трудоемкость и риски этих загрузок?

При этом, чем крупнее предприятие, чем ближе решаемые задачи к управлению ресурсами и бизнес-процессами, оперативному учету, тем специфичнее его бизнес-процессы. А соответственно, тем больше приходится дорабатывать типовое решение в ходе внедрения, каким бы хорошим и функциональным оно ни было.

Последствия некорректно прошедших обновлений приводят к ошибкам в программе, что чревато последствиями для бизнеса (остановкой и перебоями в бизнес-процессах, выдачей некорректной информации руководству и т.д.). Объем корректировок и

сверок при каждом новом обновлении работы заранее спланировать невозможно, так как неизвестно, какие изменения предложит в очередном релизе поставщик системы.

Перелом бизнес-процессов предприятия в угоду особенностям системы

Отметим, что в последнее время заказчики, опасаясь описанных выше проблем, переходят в другую крайность и требуют при внедрении минимальных доработок и использования по максимуму типовых функциональных возможностей программы. Ради того, чтобы сохранить полноценную поддержку поставщика решения, руководство предприятия зачастую бывает готово менять свои бизнес-процессы в ущерб собственно бизнесу.

Почему так происходит? Потому что, если типовое решение снимается с сопровождения, оно становится подобным кораблю, отправившемуся в вечное плавание без права пристать к берегу.

Однако такой подход неприемлем для крупных состоявшихся предприятий, бизнес-процессы которых отлажены, персонал обучен и способен эффективно работать в рамках принятых схем. Перелом бизнес-процессов, насильственное внедрение методик, диктуемых системой, может негативно сказаться на общих результатах бизнеса, нивелируя всякую ожидаемую выгоду от пользования системой.

Разумное решение — два уровня системы

Что же делать заказчику, которому необходим функциональный программный продукт, адекватно автоматизирующий существующие бизнес-процессы и при этом находящийся на поддержке у поставщика решения?

Проблема кроется в "однослойной" структуре большинства распространенных типовых решений. Единственный слой решения — это "конфигурация" или программа от поставщика решения. В одном слое переплетены процедуры, работающие со структурами данных, и сервисные процедуры, облегчающие работу пользователя. И весь этот слой сопровождается поставщиком решения. И в этом же слое (в "конфигурации") приходится делать доработки в ходе внедрения.

Решение данной проблемы простое — нужно расширить выбор типовых решений и использовать программу, имеющую двухслойную структуру. Во внешний (второй) слой кроме отчетов нужно переносить все доработки, выполненные на проекте, если эти доработки выполнены для облегчения работы пользователя. Примером таких доработок могут быть сервисы ввода данных, бизнес-правил работы системы и контроля действий пользователя, отчеты и регламентные обработки данных (например, поиск и исправление ошибок). Таким образом, вокруг основного уровня системы формируется "слой отчетов", который никак не связан с поставщиком решения и никак не затрагивается при обновлении очередным релизом от поставщика.

При таком подходе сама программа поставщика не модифицируется при внедрении, благодаря чему

новые релизы-обновления загружаются беспрепятственно одним нажатием кнопки в конфигураторе.

В идеальной ситуации все доработки, связанные с бизнес-правилами конкретного предприятия, должны находиться во внешнем слое, в виде подключаемых программных модулей. По такому принципу устроены многие западные системы: есть глубокий слой ядра, сопровождаемый поставщиком, а есть внешний слой, дорабатываемый и сопровождаемый внедренцами.

Сумма слоев обеспечивает предприятию требуемую функциональность, при этом исключается внутренний конфликт поставщика решения и внедренцев. Разумеется, программисты предприятия должны обладать достаточной квалификацией, чтобы это понимать и прикладывать максимум усилий для того, чтобы выносить доработки во внешний слой — создавать "внешние обработки".

Не все системы одинаково настраиваемы

Но насколько наиболее распространенные программы позволяют выносить бизнес-логику во внешний слой? Как в этом плане различаются разные типовые решения? В ответе на этот вопрос как раз и кроется причина, почему широко тиражируемые типовые решения снимаются с сопровождения при доработках. Потому что, как бы ни хотелось, многие бизнес-правила приходится реализовывать внутри программы поставщика (в "конфигурации").

Очень много операций бизнес-логики основаны на том, что некоторое действие пользователя должно перехватываться внешней программой и анализироваться, при этом должны выполняться некоторые процедуры, обработки согласно бизнес-логике. Кроме ввода данных в поля ввода, есть множество действий пользователей, которые надо анализировать и как-то специфически на них реагировать согласно бизнес-правилам — это открытие/закрытие форм, проведение документов, запись в БД, создание новых документов и элементов справочников, удаление данных и т.д. Не все типовые решения могут быть настроены правильным образом без изменений ядра программного продукта.

Рассмотрим простой пример. Оператор вводит типовую операцию — "приход материала на склад". При этом он заполняет документ "Поступление товаров и услуг", который содержит множество реквизитов и режимов проведения. Он должен создать документ, выбрать вид операции, заполнить необходимые настройки документа и т.д. А собственно полезная его работа заключается только в том, что он заполняет всего два реквизита — номенклатуру и количество. Все остальное за этого оператора могла бы дополнить сама программа согласно некоторым зашитым в нее правилам.

Рассмотрим, в каком случае он будет работать быстрее и совершать меньше ошибок:

а) при вводе данных в типовой документ конфигурации (не доработанной);

б) при вводе данных в специализированную форму документа, в которой есть лишь два реквизита — номенклатура и количество. Причем эта форма вся "обвешана" специфическими проверками на корректность вводимых данных.

Вариант (б) обеспечивает высокую скорость и точность выполнения бизнес-процесса, минимизирует ошибки пользователей, особенно в случае низкой компьютерной грамотности.

Как только выбирается вариант (б) как оптимальный, возникает вопрос: а можно ли эту форму (вернее, позволяет ли типовая конфигурация) вынести во внешний слой или делать ее, как дополнительную внутри типового документа?

Найти систему, решающую все проблемы, возможно

В практике авторов проходили проекты, когда из-за компьютерной неграмотности младшего производственного персонала вся конфигурация была "увешана" такими упрощенными формами (рабочими столами) с правилами проверки вводимых данных. Пользователям было запрещено вводить данные напрямую в документы, чтобы они не допускали ошибок и не влияли на данные в системе.

В результате дополнительных настроек пользователи, не обладающие даже базовым уровнем компьютерной грамотности, могли успешно эксплуатировать достаточно сложную производственную систему со сложными правилами работы. Разумеется, сотрудники бухгалтерии и менеджеры других подразделений на этом предприятии работали с типовыми документами без всяких "рабочих столов".

На описанном выше проекте использовалось типовое низкотиражное ERP-решение на платформе "1С:Предприятие 8", которое позволяло максимальный объем нетиповых функциональных возможностей вынести во внешний слой. В результате, несмотря на то, что почти все документы и многие справочники были серьезно доработаны (фактически изменился внешний вид документов), все изменения были выполнены во внешнем программном слое, а система осталась полностью на сопровождении поставщика.

Выводом данной статьи может стать рекомендация:

Типовое решение должно поддерживать возможность вынесения максимума функциональности во внешние программные модули (файлы внешних обработок), то есть поддерживать два слоя: поставщика решения (который сопровождает поставщик) и внедренца (который дорабатывают и сопровождают внедренцы). Все доработки для соответствия специфике бизнес-процессов предприятия нужно стремиться делать во внешнем слое с тем, чтобы не снимать систему с сопровождения разработчика.

Лисин Николай Геннадьевич — заместитель директора компании "Институт типовых решений — Производство" (ИТРП). Контактный телефон (495) 600-61-79. E-mail: lisin@itrp.ru