

Выбор ERP-системы: два уровня функциональности

Николай Лисин, "Институт типовых решений – Производство" (ИТРП)

Опубликовано: "Финансовая газета", №5 (февраль) 2009г.

Рассмотрим ситуацию с автоматизацией учета и управления, сложившуюся на предприятиях в конце 90-х гг. XX века, когда стали доступными различные средства создания ИТ-инфраструктуры:

- автоматизированная система предприятия включает несколько разрозненных программ и баз данных, между которыми постепенно накапливались и отлаживались силами собственного ИТ-персонала средства обмена данными;
- большая часть работающих программ - это заказные либо собственной разработки продукты или типовые решения (обычно "1С") с достаточно глубокими доработками;
- На сопровождение такой автоматизированной системы тратятся существенные силы местной команды ИТ-специалистов. Сопровождение помимо сиюминутных доработок по пожеланиям пользователей включает выполнение регламентов обслуживания этой системы, регламентов обмена, трансформации и синхронизации данных между различными базами;
- бухгалтерский учет в лучшем случае ведется с использованием типового решения. Если глубокие доработки не практикуются, существует определенная стабильность системы регламентированного учета.

В сложившейся ситуации доработки действующего программного обеспечения, как правило, не носят системный характер, поскольку направлены на удовлетворение текущих пожеланий пользователей как можно меньшими средствами. Срок функционирования таких систем не бесконечен. Наблюдается тенденция к увеличению степени проникновения решения в бизнес-процессы, что сопровождается нарастанием требований со стороны руководства и пользователей к функциональности данного программного обеспечения. Свою лепту в процессе нарастания требований вносят изменения законодательства и структуры бизнеса. Множество локальных доработок с течением времени приводят к росту сложности его дальнейшего обслуживания. Это происходит потому, что любая программа изначально проектируется в целостном виде, с продуманной архитектурой и взаимосвязями и с определенными ограничениями и допущениями, которые носят системный характер. Такую "чистую" систему дорабатывать и развивать достаточно просто. Однако в процессе дальнейшего эволюционного развития программного обеспечения сложно соблюдать "системность" доработок, так как их задачей является быстрое и эффективное решение сиюминутных задач.

При системном подходе к решению задачи скорость и эффективность всегда стоят на втором плане, поскольку основным требованием выступает необходимость учитывать свойства всей системы в целом, а не локального ее участка. Можно сказать, что решение текущей проблемы по пожеланиям пользователей должно быть таким, каким оно было заложено в систему при ее начальном проектировании в целом.

Несистемные решения (доработки) приводят к постепенному разрушению целостности действующего программного обеспечения. Это касается любого программного продукта, особенно позволяющего решать экономические задачи (учет, управление), независимо от

качества исходного решения. Фактически разрушение целостности работающего на предприятии программного обеспечения - это объективный процесс, который приводит к ограничению срока функционирования любой действующей системы автоматизации. По статистике, срок жизни автоматизированной системы, обеспечивающей решение экономических задач, - не более 5-7 лет.

Разрушение целостности информационной системы сказывается на качестве ее функционирования и ощущается пользователями. Снижаются согласованность, детальность, оперативность, достоверность получаемой информации. Развитие системы, т.е. добавление в нее новых функций, затруднено и сопровождается сопротивлением отдела ИТ. При этом опыт показывает, что программное обеспечение, основанное на платформах "1С", обладает большей степенью "живучести", так как открытость кода позволяет быстро выполнять необходимые доработки.

Оптимальное решение - внедрение нового типового продукта

Выход из ситуации видится только один - начало нового цикла развития автоматизированной системы. Новый цикл начинается, как правило, с перехода на новое типовое программное обеспечение. Лавинообразный рост числа внедрений автоматизированных систем учета и управления на небольших и средних предприятиях в начале текущего века связан с появлением платформы "1С:Предприятие 7.7". Принимая во внимание средний срок функционирования такой системы, можно сделать вывод о том, что в настоящее время на многих предприятиях ощущается острая потребность в начале нового цикла развития программного обеспечения.

Следует отметить, что типовые решения развиваются, и та функциональность, которая ранее создавалась только собственными ИТ-специалистами предприятий, появляется и в типовых (коробочных) решениях. Однако процесс их эволюции имеет важную особенность - развитие ведется системно и планомерно командой разработчиков софтверной фирмы, которые не ограничены необходимостью постоянно удовлетворять сиюминутные потребности пользователей.

Естественно, при переходе на новое программное обеспечение, руководство компании ожидает качественного улучшения функциональности автоматизированной системы. Как правило, вкладывая средства в новую систему, топ-менеджер стремится решить ключевые задачи:

- получить систему с нужной функциональностью;
- приобрести типовое решение от разработчика, нуждающегося в минимальных доработках.

Первое требование очевидно и дополнительных обоснований не требует. Но насколько справедливо ожидание получить готовое типовое решение?

Второе требование звучит все чаще, и это не удивительно - в настоящее время руководство уже способно оценить последствия постоянного вмешательства в работу системы и ее доработок. Часто со стороны заказчиков софтверных компаний выражается готовность максимальной подстройки под типовые схемы, заложенные изначально в типовом решении. В стремлении приобрести типовое решение проявляется желание руководства предприятия получить не только функциональную, но и устойчивую, стабильную систему. Сегодня устойчивость и стабильность стали одними из ключевых факторов.

Проект создания автоматизированной системы, связанный с внедрением типового решения, при ведении проекта квалифицированными специалистами, выполняется комплексно, а система изначально продумывается в целом с учетом требуемых доработок. При этом существует реальная возможность выполнить все доработки одновременно и системно. Однако практический опыт подтверждает, что любые доработки готового решения в любом случае несут в себе риски несистемности. Следовательно, чем меньше производится комплексных доработок на стадии внедрения, тем лучше это для целостности создаваемой системы, а основной риск несистемных доработок - получение тех же проблем, которые были присущи прежней системе, т.е. сложности в ее развитии и сопровождении. Именно поэтому наиболее компетентные потребители типовых решений теперь ведут тщательный анализ функциональности предлагаемых систем, запрашивают тестовые бизнес-примеры решения типовых задач.

Важно отметить, что создаваемая новая система, как бы она ни была хороша, обречена пройти тот же путь развития, что и прежняя. Срок ее существования также ограничен неизбежными последующими несистемными доработками из-за постепенного изменения бизнеса предприятия.

Критерий выбора - ранжирование требований к функциональности

При дальнейшем анализе возможностей предлагаемых систем следует учитывать, что функциональность программного обеспечения может быть поделена на два уровня:

- функциональность ядра - возможность хранить и рассчитывать необходимые данные;
- функциональность, направленная на обеспечение сервисов для выполнения бизнес-процессов и на удобство пользователей. Например, это удобные для конкретных пользователей формы отчетности, копирование данных, автоматическое заполнение документов, предоставление нужной информации в нужные моменты в удобном виде, контроль за правильностью введенных данных на этапе ввода, подсказки пользователю, дружелюбность интерфейса и т.д.

Доработки функций на уровне первой группы (функций ядра) связаны с наибольшими рисками и, как правило, влекут за собой снятие доработанных модулей с сопровождения фирмы-разработчика. Эти доработки имеют наибольший риск несистемности, и именно они наименее желательны для типового решения. Однако по этой группе функций наиболее просто определить еще на стадии выбора системы нужную функциональность. Достоверный, исчерпывающий перечень требуемой функциональности позволяет значительно снизить риск несоответствия выбранной системы ожиданиям потребителя.

Свойства и влияние на целостность автоматизированной системы доработок по второй группе диаметрально противоположны первой. Сервисные возможности по определению относятся к некоторой локальной области функций, а их модификация и доработки не разрушают функциональность ядра. Например, разработка нового удобного отчета по заявке пользователя, сервиса автозаполнения документа, сервиса утверждения документа пользователями, сервиса печати и т.д. никак не влияет на целостность системы и потенциально не способна ее нарушить.

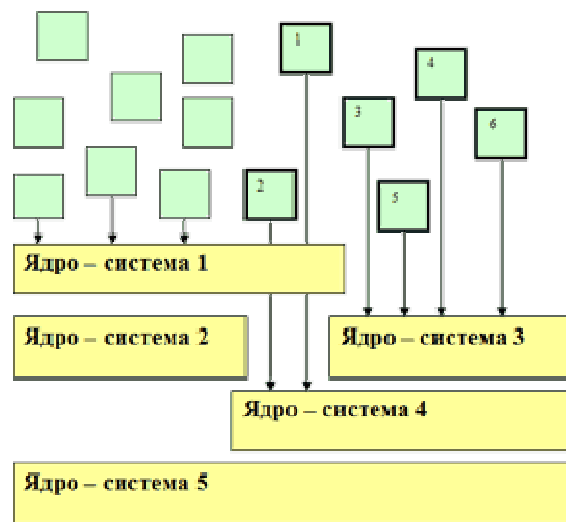
Таким образом, все требования потребителя к автоматизированной системе должны быть поделены (ранжированы) на указанные группы, а выбор системы должен быть основан на соответствии функциональности системы требованиям первой группы.

Подобный подход подразумевает, что доработки по второй группе требований обладают несущественными рисками и могут быть выполнены в зависимости от специфики предприятия. Принципиально важным является тот факт, что на второй группе требований специфика проявляется наибольшим образом, и типовое решение, как правило, эти особенности не покрывает в достаточном объеме. Вторая группа требований подвержена постоянным изменениям, и по ней ведется доработка ИТ-командой предприятия в период функционирования автоматизированной системы. Наконец, модификация в области второй группы требует меньшей квалификации программистов и в меньшей степени способна исказить данные в системе, разрушить их целостность и т.д.

Следует отметить, что изменения законодательства часто затрагивает функциональность первой группы (ядра), что приводит к необходимости обновления релизов от разработчика системы. Снятие разработчиком ядра с сопровождения влечет серьезные трудности и затраты на поддержку соответствия системы требованиям законодательства.

Итак, функциональность первой группы - это фундамент системы, вмешательство в который крайне нежелательно, функциональность второй группы - это "надстройка", которая допускает большую свободу действий со стороны потребителя, не неся при этом серьезных рисков. Разумеется, "надстройка" может существовать только на соответствующем ей фундаменте. Если в ядре нужной функциональности нет, то и о надстройке не может быть и речи.

Практика показывает, что не у всех потенциальных потребителей, выбирающих систему, есть четкое понимание указанных принципов. Рассмотрим схематично простой пример. Заказчиком сформированы на этапе выбора системы необходимость автоматизации бизнес-процессов 1, 2, 3, 4, 5, 6 (см. рисунок).



Очевидно, что потребностям заказчика удовлетворяют системы 4 и 5, поскольку имеют необходимые возможности ядра.

Неочевидно, но важно то, что системы 4 и 5 могут не иметь явно выраженной реализованной функциональности автоматизации, например, процессов 1 и 2 (либо функциональность несколько отличается от требуемой), но имеют все необходимые возможности для реализации этих функций путем доработок уже не на уровне ядра.

Далее предположим, что:

- система 3 имеет наиболее подходящую и проработанную для предприятия реализацию функций 3 и 4 (по сравнению с системой 4 и 5), но для реализации функций 1 и 2 требуется доработка ядра системы 3;
- система 4 требует некоторых доработок всех функций 1-6, но ядро не требует доработок.

В данном случае пользователь должен сам решить, какой системе отдать предпочтение - 3 или 4.

Существующие программные продукты

Практический опыт свидетельствует, что степень проработки системы на уровне сервисов бизнес-процессов (вторая группа требований) прямо влияет на ее объем, сложность настройки и стоимость как самого программного обеспечения, так и услуг по внедрению. Поэтому в равной степени на рынке существуют и объемные системы, обладающие большим ключевым функционалом на уровне ядра и предоставляющие возможности доработки функций второго уровня, и небольшие "лаконичные" решения, заточенные под задачи небольших предприятий и не требующие значимых доработок.

Рассмотрим типовые решения, поставляемые компанией "Институт типовых решений - Производство" (ИТРП), разработчиком систем управления производственными предприятиями, дочерней компанией фирмы "1С". Данные решения созданы на платформе "1С:Предприятие 8", являющейся в настоящее время одной из наиболее популярных в сфере автоматизации.

ИТРП: Процессное производство 8. Самостоятельная разработка компании ИТРП, Продвигается в отрасли процессных, непрерывных, позаказных производств. В основу концепции продукта положены практичность и технологичность, разработка и тщательная "обкатка" коробочного решения на реальных проектах автоматизации. В настоящее время это единственное тиражное решение на платформе "1С:Предприятие" со специализированной функциональностью для отраслей непрерывного производства (химическая, перерабатывающая, пищевая промышленность и т.д.).

Сильные стороны решения - проработка функционала для крупных предприятий и холдингов, наличие образцовых выполненных проектов, в т.ч. в холдинговых структурах. Важным преимуществом решения является возможность выполнять доработки функциональности группы 2 (сервисы бизнес-процессов, специфичные для предприятия) на подключаемых внешних программных модулях, без снятия с сопровождения. Для большинства предприятий это решение фактически соответствует системе 5 (иногда - системе 4), приведенной на рисунке.

"ИТРП: Производственное предприятие 8 Стандарт". Продукт представляет собой типичную легкую систему, функциональность которой ограничена четким кругом типовых задач и зачастую не подразумевает доработок какого-либо уровня, в особенности на уровне ядра системы. На рисунке выше для большинства проектов решение является системой 3.

Подобное решение идеально подходит заказчикам - небольшим производственным предприятиям, заинтересованным в оперативном внедрении при отсутствии дополнительных настроек в сочетании с разумным бюджетом проекта.

Автоматизация "под ключ" и "под развитие"

Желание решить текущие проблемы, имеющиеся в настоящее время, часто приводит заказчиков системы к требованию получить систему "под ключ". Подразумевается, что поставщик решения сначала анализирует деятельность предприятия, проектирует и моделирует бизнес-процессы, по результатам дорабатывает программное обеспечение под требования клиента, разрабатывает регламенты и обучает пользователей. Система сдается в эксплуатацию, ее последующие доработки свидетельствуют о ее некачественном проектировании. Насколько корректен такой подход, и достижимо ли требуемое для сдачи "под ключ" качество проектирования? Автоматизированная система является

динамически развивающейся. Требования, сформулированные заказчиком в определенный момент времени, могут корректироваться впоследствии бесконечное количество раз при эксплуатации системы - при переосмыслении и дополнении целей и задач предприятия, изменении законодательства и пр.

Единственно возможный критерий целевого состояния "под ключ" - отсутствие замечаний к системе у заказчика в данный момент. Однако это не означает, что система в этом состоянии удовлетворяет всем требованиям заказчика и что при ее эксплуатации у него не возникнут проблемы. Фактически принимая работу "под ключ", он принимает риски получить неудовлетворяющую его систему. Отсюда следует, что признание достижения целевого состояния "под ключ" (полной готовности системы) не в интересах заказчика. Если это признание связано с принятием работ исполнителя по договору подряда, то получается конфликтная ситуация:

- заказчику неизвестно, какие еще проблемы или пожелания пользователей могут возникнуть при эксплуатации системы, но он подозревает что их будет достаточно много. На текущий момент он не готов сформулировать претензии к системе.
- исполнитель требует принять работы, так как замечаний к системе нет.

Итак, наблюдается ситуация, когда принятая технология работы объективно сталкивает между собой заказчика и исполнителя, хотя правы обе стороны, а последствия как правило негативны для всего проекта.

Практика показывает, что если предметом договора определяется целевое состояние "под ключ" или аналогичные состояния, характеризующие "абсолютную готовую систему" (например, отсутствие замечаний в течение некоторого времени), сроки договора не соблюдаются, поскольку заказчик, как правило, не берет на себя ответственность за определение этого состояния. Требование по созданию системы "под ключ" без возможности четко определить это состояние приводит к обманутым ожиданиям и разочарованию заказчика.

Определение целевого состояния системы "под ключ" позволяет заказчику затягивать с оценкой результата и формулировать новые требования по ее доработке. При этом оценивать его действия как недобросовестные не всегда правильно, так как он защищает свои интересы исходя из условий договора и принятой сторонами технологии проектных работ.

Заказчик "защищает" свои интересы в сложившейся ситуации следующим образом. В процессе оценки результата работы "А" (этот процесс занимает некоторое время) заказчик обнаруживает необходимость доработки в выполненной работе "В", исполнитель начинает эту доработку. На тот момент, когда результат работы "А" будет оценен, работа "В" еще не будет выполнена или выполнена, но заказчик будет занят оценкой ее результата. Когда работа "В" будет оценена, исполнитель будет загружен работой "С" и т.д.

Таким образом, целевое состояние, при котором замечания заказчика отсутствуют, не достигается в силу вполне обоснованных объективных причин. Это позволяет заказчику совершенно обоснованно не принимать результат работы по договору подряда и снимает ответственность с его конкретных представителей за приемку работ. В то время, когда система на 95% работает и удовлетворяет потребностям заказчика, работы исполнителя не будут приняты.

Подытожим риски исполнителя при согласии принять на себя обязательства по сдаче системы "под ключ":

- система удовлетворяет требованиям заказчика на 95%, используется в промышленной эксплуатации;
- работа по договору не принята, следовательно, заказчик имеет рычаги давления на исполнителя;
- конкретные представители заказчика не несут ответственность за приемку работ, так как работы не приняты;
- исполнитель бесплатно в течение неограниченного времени занимается внесением изменений в систему в соответствии с очередными требованиями заказчика, которые возникают у последнего в процессе эксплуатации системы, т.е. фактически занимается сопровождением системы.

Нужно заметить, что компетентность заказчиков от крупных предприятий в последние годы существенно возросла, и с требованиями решить проблемы путем создания системы "под ключ" исполнителю приходится сталкиваться достаточно редко.

На стадии опытной эксплуатации происходит привязка системы к реальной жизни. Практика вносит неизбежные коррективы в теоретически построенную систему, исполнитель совместно с заказчиком ищет оптимальные пути решения не учтенных при проектировании проблем, заказчик может переосмыслить некоторые задачи и даже цели проекта.

На практике предварительная оценка заказчиком результата и выдача исполнителю основной массы требований по доработке происходит во время опытной эксплуатации, остальные требования заказчик выдает исполнителю в течение нескольких лет, практически - на протяжении всего жизненного цикла системы. Если система спроектирована и сделана изначально качественно, если все ключевые требования заключены в неизменном ядре системы, то объем доработок системы по второму уровню требований в единицу времени быстро снижается в процессе промышленной эксплуатации.

Признание сторонами очевидного факта, что система должна постоянно развиваться и совершенствоваться в процессе эксплуатации и это развитие является составляющим ее жизненного цикла, означает, что заказчик признает возможность запуска системы в опытную эксплуатацию в "сыром" виде - только по результатам предварительных испытаний. Стороны приходят к соглашению, что в опытную эксплуатацию продукт принимается в состоянии "под развитие", т.е. прототип системы. При этом гарантией необходимого качества системы на этом этапе являются результаты предварительных испытаний по утвержденному заказчиком сценарию испытаний.

Отметим, что испытания системы перед опытной эксплуатацией способны выявить только часть недостатков. Доведение ее до нужных параметров качества и соответствия бизнес-процессам происходит в процессе "обкатки" - опытной эксплуатации.

Заключение

Ключевым фактором в выборе решения, которое ляжет в основу автоматизированной системы предприятия, является определение требований к функциональности системы. Часть основополагающих требований должна быть изначально реализована на уровне ядра системы, которое впоследствии не будет подвергаться доработкам в течение всего

жизненного цикла системы (5-7 лет). Требования к функциям, заточенным под особенности конкретного предприятия (второй уровень требований), реализуются на этапе проектирования и подготовки системы к запуску. Но по мере развития предприятия, корректив законодательства, перемен в рыночной ситуации изменяются и эти требования, следовательно, доработки второго уровня выполняются в процессе функционирования системы.

Преимущества технологии сдачи системы "под развитие" позволяют снять потенциальный конфликт между заказчиком и исполнителем, справедливо распределив риски внедрения и обеспечив нормальные отношения, способствующие успеху проекта.

Н. ЛИСИН

Компания "Институт типовых решений - Производство" (ИТРП)