

14 типовых ошибок и проблем на проектах автоматизации

Примеры и описания ситуаций с проектов внедрения систем на
платформе «1С:Предприятие»



Авторы: А.Вепринцев, Н.Лисин.

Любое использование материалов книги возможно только по согласованию с
фирмой ИТРП (www.itrp.ru, info@itrp.ru)

Москва, 2012 г

Перечисленные проектные проблемы не являются самыми часто встречаемыми. И не обязательно именно эти проблемы самые критичные.

Этими набросками мы хотим очередной раз привлечь внимание к важности технологичного подхода к проектам автоматизации и к тому, что никогда не бывает лишним внимательный анализ опыта - успешного и неуспешного, своего и чужого...

К каким финансовым потерям может привести хотя бы одна из этих проблем, возникшая именно на вашем проекте автоматизации? Нередко эти убытки могут превысить ожидаемый эффект от результата проекта... Посмотрите, этих проблем не так уж сложно избежать. Достаточно всего лишь заранее о них помнить и предусматривать способы их решения.

Оглавление

Проблема 1. Отказ от проектного подхода там, где он был необходим	3
Проблема 2. Невнятные цели проекта	4
Проблема 3. Слабое вовлечение исполнителей в планирование	6
Проблема 4. Крайности в формализации – недостаточность и избыточность	7
Проблема 5. Неподходящий менеджер проекта	9
Проблема 6. Нерелевантные нотации функционального моделирования	12
Проблема 7. «Функциональное моделирование» без моделирования	13
Проблема 8. Затянутая стадия реализации.....	16
Проблема 9. Недостаточный оперативный контроль.....	17
Проблема 10. Растущие «междуэтапные» интервалы на сдачу-приемку	19
Проблема 11. Недостаточное внимание к управлению рисками	21
Проблема 12. Проблемы при выборе типового решения.....	22
Проблема 13. Полное непонимание политической ситуации	25
Проблема 14. Иллюзия о невозможности остановить проект.....	27
Что делать с этими проблемами?	28

Проблема 1. Отказ от проектного подхода там, где он был необходим

Бывает так, что интегратор навязывает клиенту проект в ситуации, когда предприятию – крохотной фирме - нужны совсем небольшие доработки типового решения и мини-обучение. Надо заметить, что бывает это редко. Потому что клиент хорошо понимает, что заплатить придется существенно больше, чем за несколько консультаций и обосновать необходимость проекта здесь просто невозможно. Но даже если предприятие согласилось на проект – проблемой будет только существенные затраты, при том что результат скорее всего будет успешно достигнут (а это все же основной критерий успешности сотрудничества клиента и интегратора).

Гораздо чаще встречается обратная ситуация – когда задачи автоматизации нетривиальны, подход «программирования на коленке» влечет огромные риски, предприятие хочет сэкономить и отказывается от проекта, а партнер идет у заказчика на поводу и соглашается работать на условиях почасовой оплаты без проектного подхода.

И эта ситуация хуже предыдущей тем, что можно потратить время и деньги, а приемлемый результат так и не будет достигнут.

В чем принципиальное отличие проектного подхода? На проекте составляется ТЗ? Нет, это совсем не главный критерий. Из академического определения, проект – это разовая деятельность с заведомо определенными целями и результатом. Таким образом, подход, когда работы разбиваются на стадии, по каждой из которых в явном виде обозначены требования к результату, существенно снижает риски недостижения общей цели, ради которой и затевалась автоматизация (про цели еще далее поговорим отдельно).

У нас не так давно стартовал проект, на первый взгляд состав задач представлялся несложным – внедрение регламентированного учета на небольшом предприятии, всего не более 10 автоматизированных рабочих мест. Начался он именно как проект, но в ходе выполнения первой стадии (функциональное моделирование) заказчик начал настаивать, чтобы сразу приступали к эксплуатации системы с одновременным выполнением настроек. Одной из причин такого ухода от проекта было то, что ему было сложно вникать в функциональную модель и не очень хотелось прилагать для этого усилия.

И мы согласились, на свою голову.

Уже в процессе программирования выявился целый ряд особенностей учета на этом предприятии. Основной вид деятельности заказчика – выполнение подрядных работ с использованием своей техники и рабочих, по этим работам необходимо определять нормативные и фактические затраты. Причем алгоритм расчета затрат крайне сложен, зависит от множества параметров, часть которых менеджеры предприятия не могут сходу вспомнить и выявляют уже после завершения очередной итерации программирования и

ввода данных. Сроки настройки программы затянулись в разы, в итоге мы получили один из редких неуспешных проектов. Конечно, продолжая итерационно дорабатывать систему, мы в конце концов реализовали бы все требования всех пользователей, но многократное увеличение сроков стало уже неприемлимым и для нас, и в большей степени – для заказчика. Повторюсь, на первый взгляд казалось – ну что там сложного может быть в регламентированном учете, мы же делали это уже не один десяток раз!

Так когда можно отказаться от проектного подхода, а когда лучше отказаться от клиента, чем ввязаться в авантюру затяжных настроек системы без специфицированных требований? Конечно, в решении этого вопроса требуется опыт экспертов и четко рационального ответа быть не может. Но в общем случае, если есть хотя бы небольшие сомнения, то лучше не рисковать и следовать проектной технологии.

Проблема 2. Невнятные цели проекта

Раньше на наших курсах по проектным технологиям мы уделяли особое место целеполаганию на проекте. Почему? Потому что в то время мало кто из партнеров об этом задумывался и редко где об этом упоминалось. Сейчас про «правильные» цели можно увидеть в самых разных источниках и на многочисленных курсах. Так что здесь не будем подробно останавливаться на этом вопросе, но несколько слов все же сказать надо.

В отличие от гуру управления проектами, я бы не стал излишне драматизировать: «Без конкретных измеримых целей проект обречен!» - в нашем случае это некоторое преувеличение...

Все-таки масштабы проектов автоматизации не столь значительны, как, скажем, проект разработки нового двигателя для болида «Формулы-1» (там действительно должны быть в явном виде специфицированы все целевые показатели). В большинстве случаев цели наших проектов или очевидны всем участникам или хотя бы проговорены и не меняются в ходе работ. Тем не менее, иногда на себе можно прочувствовать особые, сложные случаи.

Далее случай из практики. Предприятие по производству и продаже светотехники поставило задачу централизовать оперативную информацию от многочисленных филиалов – управление производством, закупки, продажи, взаиморасчеты. В ходе экспресс-обследования эта задача уточнялась, но конкретные и измеримые цели проекта никто не формулировал. В общем виде цели понимались так, что проект был затеян для повышения точности данных от филиалов – на начало проекта информацию собирали кто как мог и ее достоверность была очень низкой. По прошествии нескольких стадий проекта, уже после реализации программного кода настроек приходит менеджер заказчика и говорит:

- Дорогие друзья, а почему это у нас перепроведение документов так долго проходит?
- Каких таких документов? – спрашивает наш РП.
- Ну как, разрешения на отгрузку, по складу №4.
- Где же долго, когда полный комплект по одному заказу проводится меньше 10 минут? Там один документ за собой еще несколько тянет. Для такого сопутствующего расчета это нормально, а не долго.
- В общем случае может для других ваших проектов и нормально. Но обработка именно этих документов для нас критична, - возражает заказчик. - На поступающие данные нам необходимо реагировать незамедлительно, практически в реальном времени. У нас клиент стоит у стола менеджера и ждет, будет ли ему сейчас отгрузка разрешена или с его взаиморасчетами и резервами есть какие проблемы. И по 10 минут висеть над душой у менеджера и ждать пока проведется документ - никуда не годится. А если клиент не один, а два заказа оформлял?

Итак, выясняется, что для заказчика очень важным является не только наличие централизованной информации, но и скорость ее обработки. Это действительно было критично для бизнеса – помогало сократить простои в складских операциях при приемке ТМЦ. А проведение данных не по одному, а по всем филиалам занимало больше часа, что оказалось совершенно неприемлемым. Пришлось понести дополнительные трудозатраты для оптимизации по скорости кода всех документов, связанных со взаиморасчетами.

При чем тут целеполагание?

Дело в том, что если бы целевой показатель скорости обработки был декларирован (а он **мог быть** декларирован, так как важен для бизнеса), то действительно далее мы могли бы сами предвидеть и отметить это требование в ТЗ. А на следующей стадии программисты обязательно обращали бы внимание на время проведения накладных и документов движения денежных средств.

Известная рекомендация, о которой многим хорошо известно: цели должны быть СМАРТуемы. То есть соответствовать критериям аббревиатуры SMART (specific, measurable, achievable, relevant, timed):

- конкретными – однозначно понимаемыми;
- измеримыми – как мы узнаем, что цель достигнута?
- достижимыми – проект действительно может решить эту проблему;
- релевантными – цель достигается именно проектом автоматизации, а не каким-то совсем другим способом;
- определенными во времени – когда предполагаем достигнуть цели?

Другая известная (но в меньшей степени) рекомендация по целеполаганию в области автоматизации: цель должна быть внешняя по отношению к проекту. Цель должна отвечать на вопрос, какие преимущества получает бизнес заказчика от нашего проекта?

Пример некорректной (НЕ-внешней) цели: создать автоматизированную систему поддержки формирования регламентированной отчетности. Из формулировки непонятно, а чем предприятию плохо в текущей ситуации, зачем ему нужна программа? Пример корректной формулировки: уменьшить трудоемкость формирования регламентированной отчетности с помощью системы, сократив время подготовки баланса с двух недель до одного дня.

- *Куда ты хочешь прийти?*

- *Не знаю, мне почти все равно – начала Алиса...*

- *Тогда все равно куда идти – ответил Кот.. (Л.Кэррол).*

Целеполагание – не панацея и не самый важный пункт на проекте автоматизации. Но основная идея тут в том, что определить их не является сверхтрудоемкой задачей, зато польза вполне может быть не эфемерной, а реальной. Цели задают **вектор**, направление «думания» для всей проектной команды.

Проблема 3. Слабое вовлечение исполнителей в планирование

Один из замечательных способов сделать проект безнадежным и отправить проектную команду по так называемому «пути камикадзе» - это никогда не спрашивать непосредственных исполнителей об их оценках трудозатрат по их же задачам. Директивно поставленный срок (или директивное «продавливание» срока, фактически вымучивание от исполнителя согласия взять на себя нереальное обязательство) с большой вероятностью сделает проект неуспешным.

Понятно, что на производстве, когда токарю Василию Федоровичу ставится задача выточить десять совершенно типовых штуцеров, начальник цеха редко интересуется:

- А что, Федорыч, сделаешь эту работу за день? Или за неделю? Или может тебе помощник нужен?

Есть норматив на типовую операцию. Поэтому всем и так понятно, за какое время в соответствии с нормативом рабочий должен выполнить задачу. С проектами автоматизации несколько сложнее...

Вообще, для ИТ-проектов основными способами оценки трудозатрат для последующего планирования являются нормативный способ и метод экспертно-статистической оценки. В фирме ИТРП мы не пробовали применять нормативный способ в полном объеме, на мой взгляд разнообразие задач столь велико, что их стандартизация вряд ли возможна. Я слышал, что некоторые партнеры пробовали составлять специальные нормативы на программирование, но о действительно успешных результатах мне не известно. Хотя на поверхностном уровне здравого смысла нормирование все же имеет место.

Например, функциональное моделирование бизнес-процесса «Управление производством» может занять от недели в простейшем случае до месяца в наиболее сложном, а на создание печатной формы документа «Заказ на производство» требуется несколько часов, за исключением совсем нетиповых форм с выводом сложно рассчитываемых данных. Таким образом, если консультант берется провести полноценное моделирование процесса производственного планирования, с требуемым вовлечением менеджеров предприятия, за один день – вероятно, он недооценивает задачу.

*Аналогично, если программист считает, что на разработку печатной формы, скажем, наряд-заказа ему потребуется неделя, то руководителю проекта надо вдумчиво проанализировать **вместе** с программистом – почему он так считает.*

Метод экспертно-статистической оценки заключается в том, что оценки трудоемкости задачи (в выражении требуемого на нее времени) опрашиваются те, кто делал нечто хотя бы примерно подобное; затем оценка усредняется. Не обязательно руководитель проекта является самым квалифицированным и знающим экспертом в области прогнозирования трудозатрат. Не обязательно таким экспертом является и сам исполнитель задачи – но в любом случае при определении продолжительности как минимум необходимо учитывать оценки этих двух участников команды. Получится получить оценку еще от кого-то более опытного – отлично, но это уже не самоцель.

И еще раз отметим, что полное игнорирование мнения исполнителя задачи встречается довольно редко, а вот неосознанное или осознанное прогибание сотрудника руководителем на взятие невыполнимого обязательства по сроку – бывает сплошь и рядом. Будьте поаккуратнее с силовым воздействием: как гласит закон термодинамики Мерфи, «под давлением все ухудшается»!

Проблема 4. Крайности в формализации – недостаточность и избыточность

Очевидно, на проекте нужна формализация задач. Как и во многих других областях, следует выбирать «золотую середину» при установлении степени этой формализации. Хорошо известно, что наиболее весомым риском является не избыточная, а недостаточная документированность задач. И реализация этих рисков происходит гораздо чаще. Акцент именно на недостаточности формализации обусловлен в том числе и тем, что чрезмерная документированность обычно не сказывается отрицательно на результате работ. А скорее даже наоборот – способствует достижению этого результата. Вопрос только в том, какой ценой этой происходит? После определенной точки риски снижаются до минимально разумного уровня и каждый последующий день работы над детализацией требований является неоправданным с позиции соотношения «результат/затраты».



В прошлом году мы выполняли проект по автоматизации бухгалтерского и налогового учета у производителя смазочных материалов. Основной объем работ был связан с выверкой данных, с переносом их из старой системы в новую. Предметная область не предполагает особенных нетиповых изысков, поэтому формализация требований была минимальная. На начальных стадиях наши эксперты проявили высокую компетентность, поэтому с заказчиком сложились хорошие, доверительные отношения. Через некоторое время формализация задач почти прекратилась полностью.

Надо отметить, что в таком крайнем проявлении проблема недостатка формализации начинает перекликаться с отмеченной ранее Проблемой 1 – отступление от проектного подхода.

Поскольку по определению вехи проекта должны быть критериальными, а если критерии вообще никак не формализованы – то какая же это критериальность? Все бы хорошо, но через 3-4 месяца заказчик начал волноваться: вроде мы что-то делаем, но то ли делается, что действительно необходимо в первую очередь? Минимальное оформление списка задач в виде спецификации требований к ключевым результатам (которые надо достигнуть в текущем месяце) сразу возымело положительный эффект, проект вернулся на нормальные рельсы, взаимное доверие восстановилось.

Другой пример. Фирме ИТРП была поставлена задача автоматизация всех основных бизнес-процессов на крупнейшем производстве строительных материалов. Причем в связи с завершающейся реструктуризацией предприятия был крайне важен срок окончания проекта – новые созданные структурные единицы должны были начать работу уже в единой информационной системе. Для того, чтобы обеспечить еще более строгий контроль над процессом автоматизации, заказчик привлекал для управления проектом со своей стороны внешних (аффилированных) консультантов. С их стороны было обозначено пожелание формализовать все что можно (и что нельзя):

- А давайте вы в ТЗ не только требования к функциям укажите, но и пропишете все реквизиты и формы документов, ага? И диаграмму бизнес-процессов надо исчерпывающую нарисовать, до уровня исполнения функции бригадой кирпичного цеха! Вы же сможете подробно перечислить в каждой функции, какие бумажки сейчас бригадир заполняет при выпуске каждой вагонетки готовой продукции?

Мы, конечно, смогли бы все эти пожелания удовлетворить. Но, напомним, срок сдачи проекта был крайне критичен, а с такой высокой степенью формализации мы бы к плановой дате завершения как раз успели бы только проектную документацию разработать. Поэтому с заказчиком и с консультантами провели разъяснения:

- Требования к функциям мы обязательно формализовываем, модель бизнес-процессов тоже обязательно составляем! Но уж способы реализации документов (если они не принципиальны для выполнения бизнес-функции) оставляем на свое усмотрение. Иначе в плановый срок уложиться не реально.

Все поняли, согласились. Выбранный уровень документирования оказался полностью адекватным (и не недостаточным, и не избыточным) – система была успешно введена в эксплуатацию через 5 месяцев с начала работ, причем в соответствии с исходными требованиями к ее функциям.

Действительно, разумно ли использовать формализационные подходы от проектирования космического корабля на тривиальном бытовом проекте замены лампочки в люстре?..

Проблема 5. Неподходящий менеджер проекта

Думаю, ни у кого нет сомнений в том, насколько сильно зависит успешность проекта от личности Руководителя проекта (РП). В нашей практике было несколько случаев, когда гениальный РП вытаскивал крайне сложные проекты из казавшихся тупиковыми ситуаций. А бывало и так, что ситуация на образцово-показательных проектах начинала ухудшаться и проблемы нарастали, как снежный ком – именно из-за плохого управления.

Не будем здесь подробно перечислять требования к личностным и квалификационным качествам РП – развернутые списки и описания легко найти в каждой второй книжке по управлению проектами. Для себя, в компании ИТРП, мы просто придерживаемся нехитрого заключения: менеджер проекта обязательно должен иметь технические компетенции и хорошо разбираться в архитектуре системы, но еще более важным требованием является наличие администраторских способностей. Еще раз: важно и то, и

другое, но менеджерские навыки – чуть приоритетнее. Сказанное относится и к назначению руководителя проекта со стороны предприятия (неважно, привлекаются ли при этом внешние подрядчики или нет) – последствия ошибочного назначения примерно одинаковы.

Одно время мы пробовали разделять роли архитектора системы (в некоторых компаниях эту роль называют *Технический руководитель проекта*) и РП как менеджера – только администратора.

В смысле, пробовали разделять эти функции между разными людьми, назначая на проект отдельно Архитектора системы (ответственного за ключевые технические решения) и РП (ответственного только за администрирование).

Возможно на каких-то мегакорпоративных суперпроектах это оправдано. Но даже на наших довольно крупных работах по внедрению «1С:Предприятие 8» практика показала, что совмещение этих функций в одном лице – гораздо предпочтительнее. Да, конечно – человека, удовлетворяющего сразу двум классам требований, найти сложнее. Но усилия по преодолению этой сложности, и как следствие, назначение на проект такого РП всегда оказываются обоснованными.

Вот пример, когда руководителем проекта был назначен менеджер-администратор, без каких-либо технических компетенций. Он управлял проектом на крупном предприятии химической промышленности. Сотрудник недавно закончил курсы по управлению проектами (вроде бы даже не один курс, а несколько) и начал успешно применять свои познания на практике. При этом в команде присутствовали технически подкованные специалисты. Сразу скажу, что ничего страшного на этих проектах не случилось.

Все-таки строгое следование технологии – это великое дело!

В многочисленных бумажках у этого РП был безукоризненный порядок, ключевые вопросы проекта всегда протоколировались, еженедельно предоставлялась отчетность, все регламентирующие документы всегда были вовремя оформлены. Однако хотя он и стремился отодвинуться подальше от технических вопросов и заниматься только управлением – иногда приходилось все-таки вовлекаться в технологические процессы.

И тут во всех красках проявлялась его недостаточная техническая компетентность. В таких условиях руководитель проекта не мог сохранять надлежащий авторитет, ни перед заказчиком, ни в собственной команде. Это усложняло коммуникации в команде, требовало более формальных подходов. Росли трудозатраты на усложненные процессы согласований ключевых технических решений. Отсутствие авторитета у РП подрывало доверие между участниками, которое, как известно, является катализатором любого проекта. В результате проект держался на плаву, но держался на бюрократических

подпорках, замедляющих движение (ничего не поделаешь – если упразднили бы бюрократию, то проект бы вообще обрушился).

Через полгода после начала проекта мы все же заменили РП. Надо еще раз отдать должное следованию технологии и методикам управления проектами - все это снизило «привязанность» работ к персоналиям и замену удалось провести относительно легко. Возглавил команду эксперт, являющийся хорошим менеджером и при этом технически грамотный. Скорость работы возросла в разы, буквально за месяц проект был успешно завершен.

Гораздо хуже, когда на роль руководителя проекта назначается технически компетентный консультант, возможно даже знающий принципы управления проектами, но не обладающий организаторскими способностями. Когда мы один или два раза допускали такую ошибку в выборе менеджера проекта – последствия неизменно оказывались проблемными.

Подобный пример был около трех лет назад на автоматизации предприятия, производящего упаковку. Проект намечался относительно несложный и включал в себя задачи только производственного и регламентированного учета. Поэтому мы сочли возможным назначить руководителем эксперта, досконально разбирающегося в этих предметных областях, отлично знающего типовое решение, но обладающего малым опытом организации и контроля проектных работ. Надо сказать, что на начальных этапах мы получали от руководства предприятия положительные отклики:

- Спасибо, мы очень довольны руководителем проекта! Он очень компетентный, разговаривает в терминах предметной области, понимает наши требования с полуслова, хорошо их формализовывает!

Но через некоторое время мы заметили периодические отклонения от технологии, проявляющиеся все чаще, пользователи предприятия начали продавливать неадекватные методологические решения, формализация стремительно сократилась почти до нуля. Мнение заказчика о проектной команде стремительно поменялось:

- Что нам тут напрограммировали в программе? Мы же просили другое! А вы наши требования почему-то совсем перестали записывать!

Проблема была не столько в том, что РП не знал, как управлять – он был прекрасно знаком со всеми концепциями управления проектами. Дело в другом – ему не хватало жесткости и воли удерживать работы в рамках технологии. Проект стал полностью неуправляемым, про цели автоматизации все забыли, руководитель проекта начал метаться, дергая программистов и «на коленке» реализовывая пожелания со слов пользователей... На приемлемый уровень удалось выйти только после подключения к проекту нашего дополнительного эксперта-куратора.

В существовавшей некоторое время назад ситуации кадрового голода иногда приходилось идти на компромиссы при назначении РП. Но сейчас, когда реально есть возможность выбирать – подходите к назначению осмотрительно.

Проблема 6. Нерелевантные нотации функционального моделирования

На проводимых нами курсах по проектным технологиям слушатели периодически задают вопрос: а какие нотации лучше использовать для изображения схем и моделей бизнес-процессов? В ответ мы все время обращаем внимание на следующее: на этой стадии гораздо важнее, что и как моделировать, а уж в каком формате схему нарисовать – дело далеко не первостепенной важности.

Функциональное моделирование – это анализ бизнес-процессов, подлежащих автоматизации. Причем анализ не должен быть оторван ни от реалий предприятия, ни от прототипа типового решения – будущие ключевые пользователи (менеджеры) должны быть вовлечены в процесс рассмотрения их будущей работы с системой и, по возможности, это рассмотрение должно выполняться с максимальным использованием типового решения, демонстрации его возможностей, с попытками выполнить с его помощью элементы типовых бизнес-операций.

Надо сразу же по максимуму привлекать всех будущих пользователей к моделированию:

- Ага, сейчас вот эти параметры учитываются у вас в бумажном журнале так? А вот это – в электронных таблицах, верно? Давайте вместе посмотрим, как это можно будет делать в программе. Попробуйте сами вводить вот в этот электронный документ те данные, которые вносили ранее в электронную таблицу. В какой момент вы должны ввести документ? (какое событие инициирует ввод?) От кого получаете необходимую информацию? С какой первички? Похоже, в типовом решении не хватает вот такого-то разреза аналитики, который вам необходим, верно? Ок, давайте все это и зафиксируем!

Итак, если есть хоть какая-то возможность в самом начале работ дать пользователю «пощупать» будущую систему – этой возможностью нельзя пренебрегать.

Соответственно, нотация, в которой мы фиксируем результат сопоставления пользователем его текущей работы с информацией и того, как он это будет делать в дальнейшем – должна быть максимально проста и понятна для понимания.

На автоматизации одного крупного машиностроительного предприятия мы работали вместе с партнерами. Их консультанты были весьма квалифицированы и продвинуты. И был у них особый корпоративный стандарт – обязательно составлять подробнейшую схему бизнес-процессов «как есть» и «как должно быть» в нотации IDEF0. Нет, я лично против этого формата ничего не имею: он своими требованиями отлично структурирует мышление и заведомо помогает избежать ряда ошибок в описании. И сама концепция

моделирования «сверху-вниз», заложенная в IDEF0, не позволяет упустить при детализации важные для предприятия функции. Но как же тяжело было разбираться в этих схемах заказчику!

Принесли мы коммерческому директору на согласование наши модели. Он только что поругался на своих снабженцев, да еще с поставщиками по телефону поспорил – сидит весь красный, лоб утирает.

К нам он прекрасно относится, поэтому честно пытается вникнуть в многочисленные листы диаграмм, на которых функции коммерческих служб последовательно детализируются, детализируются, детализируются...

Однако на десятом листе коммерческий директор уже заметно сник, несмотря на то, что мы старались по ходу объяснять суть изображенных функций. Долистал до конца из вежливости и говорит: «Да! Все правильно вы здесь нарисовали!». Но разве такое формальное согласие нам нужно? А если в модели что-то пропустили, ведь на следующих стадиях гораздо сложнее будет это учесть!

Нотация должна соответствовать контексту и сложности задачи. Консалтинг по полному реинжинирингу бизнеса удобно (наверное?) проводить с использованием моделей IDEF0. А для изображения информационных потоков в контексте проекта автоматизации – на наш взгляд, вполне достаточно более простых форматов. Например, нотации, соответствующей встроенному механизму бизнес-процессов в платформе «1С:Предприятие 8». Нотация АРИС тоже достаточно понятна и наглядна (речь не о программном продукте ARIS Toolset, а именно о формате изображения схем) – мы часто ее используем.

Проблема 7. «Функциональное моделирование» без моделирования

Чуть выше говорилось о том, что является основой содержания работ на стадии функционального моделирования. Было удивительно обнаружить, что многие наши партнеры напрочь об этом забывали. Причем, что интересно, чаще всего забывали именно наиболее опытные партнеры и их опытные консультанты, которые занимались автоматизацией не два и не три года, а гораздо больше. То есть имеют огромный стаж работы.

Оказалось, что в этом есть некоторая закономерность. На начальной стадии моделирования зачастую занимаются «не тем, чем надо» специалисты, впитавшие в себя практику выполнения начальных стадий при создании заказных систем «с нуля» (обычно такие системы, без использования типовых решений, создавались около десяти лет назад – сейчас это редкость). В те давние времена акценты на первых стадиях проекта (обычно это называлось «предпроектное обследование») ставились в первую очередь на том, чтобы исчерпывающе описать существующие – до автоматизации – бизнес-процессы

предприятия. То есть, если у нас нет готового прототипа (Типового Решения), тогда нет и эталонных методик. Соответственно, чтобы построить модель «как должно быть» (то есть как должны выполняться бизнес-функции при их автоматизации) нужна была некая отправная точка - и таковой точкой являлось описание «как есть».

Сейчас все по-другому. Напомним, при использовании типового решения основным результатом начальной стадии проекта является моделирование в типовом решении всех основных операций учетно-управленческой деятельности.

А вот какой полезный отрицательный опыт был у нас пару лет назад на небольшом приборостроительном предприятии.

Полезный – потому что хоть мы и стараемся обучать сотрудников на чужом опыте, но обучение на своем опыте не менее, а то и более действенно!

Консультант, выполнявший проект, имел огромный стаж работы, однако в нашей компании работал не очень давно. В используемых им проектных методиках глубоко укоренились устаревшие подходы – детальнейшим образом описать работу предприятия, как она выполняется сейчас. Именно этим он и занялся на стадии функционального моделирования (надо признать, при некотором попустительстве руководителя проекта). И выполняемые на заводе операции были им описаны безупречно! Только как раз этого от него не требовалось, раздел «как есть» предполагал краткое, поверхностное отображение основных **особенностей** процессов.

Кстати, в современной технологии мы в большинстве случаев этот раздел вообще опускаем.

Самое главное, что ожидалось от консультанта на начальной стадии – вовлечение менеджеров предприятия в процесс моделирования их работы на типовом решении. Но на это ему по понятным причинам не хватило времени. Поэтому раздел «как должно быть» он писал без участия представителей завода, то есть без будущих пользователей: в академических условиях, в кабинетной тиши офиса... Получилось «моделирование без моделирования», так как при всей его опытности, изложение собственных идей по использованию методик типового решения полноценным моделированием, строго говоря, не является.

Что в итоге? Нет, катастрофы не случилось. Все-таки раздел «как должно быть» в отчете присутствовал, хоть и в «скомканном» варианте, ввиду нехватки времени. Поскольку консультант, как мы говорили, обладал значительным опытом, идеи в модели «как должно быть» изложены по большей части разумные, хотя и не безупречные. Но во-первых, заказчик с полным на то основанием выразил недоумение насчет перекоса в результатах работы: «Зачем такой фундаментальный труд в части описания, как бизнес-

операции у нас делаются сейчас? При том что описание того, как оно будет делаться в системе – настолько куцое!»

Удачную метафору на эту тему как-то привел ИТ-директор совсем другого предприятия (крупного металлургического холдинга): "Зачем мне эти описания исходного состояния, если вы все хотите делать по-другому? Я готов предостеречь своих коллег от лишних трат, потому что это пустая работа. Если вы знаете, что собираетесь сносить здание, зачем нам его схема или чертеж перекрытий?"

Во-вторых, сдача-приемка функциональной модели оказалась крайне затруднена и затянута на длительный срок по той причине, что пользователям очень сложно вникать в идеи, к генерации которых они были слабо причастны. Действительно, когда на примере типового решения они сами вместе с консультантом попробовали выполнить свои основные операции учета и управления, то достигается полное понимание, как они дальше будут вести свою деятельность с помощью автоматизированной системы. Для менеджеров предприятия становится вполне ясно, что вот такая-то бизнес-функция ложится на систему «один в один», в регламент вот этой бизнес-функции придется внести небольшие коррективы, а вот здесь будет настроено типовое решение...

При этом документ «Функциональная модель» фактически просто фиксирует все то, что пользователи «пощупали вживую» вместе с консультантом – им остается только просмотреть модель и убедиться, что зафиксировано оно без ошибок.

А при отрыве менеджеров и пользователей от процесса моделирования Отчет для них становится стопкой бумаги (да еще и написанной на «птичьем языке»). Чтобы разобраться в нем, заказчику придется самостоятельно анализировать и в уме моделировать предложенные концепции – это очень и очень непросто, особенно учитывая то, что заказчик вовсе не обязан досконально знать эталонные методики типового решения.

Вообще, стадия функционального моделирования весьма важная, поэтому в нашем дистанционном тренинге мы стараемся рассматривать ее довольно подробно: предлагаем практические задания, приводим ролики и видеозаписи и т.д. Тем самым в рамках тренинга мы детализируем и раскрываем поставленный здесь акцент:

Ценность рассматриваемой стадии – именно в опробовании типового решения на примерах, обязательно с полным вовлечением представителей предприятия в этот процесс!

Проблема 8. Затянутая стадия реализации

Для любой из стадий проекта желательно ориентироваться на ее продолжительность (или на продолжительность каждого ее подэтапа) около одного месяца. Можно было бы разбить работы еще мельче, на много-много совсем коротеньких этапов – но их же надо сдавать! И на сдачу-приемку каждого такого крохотного этапа в итоге можно потратить больше времени, чем на сами работы. С другой стороны, длительные стадии и этапы плохи тем, что в бизнесе предприятия могут происходить перемены – и не исключено, что важные изменения произойдут в процессе затяжной стадии, а мы об этом узнаем только тогда, когда придет время сдавать результат. Есть и ряд других причин...

Затянутые начальные стадии нежелательны, но обычно не слишком опасны. В начале проекта можно позволить исполнителям демонстрировать результаты через продолжительное время – у всей проектной команды, в том числе со стороны предприятия, есть некий «резерв терпения». Если в бизнес-целях зреют какие-то изменения – то как раз на начальных этапах они выявляются.

Поэтому вряд ли случится так, что изменения в приоритетах станут полным сюрпризом после завершения одной из начальных стадий.

При функциональном моделировании подразделения предприятия полностью вовлечены в этот процесс и хорошо себе представляют, что именно мы вместе с ними сейчас делаем – ни у кого не складывается впечатления, что проектная команда занимается неизвестно чем. Затяжная стадия реализации настроек в этом смысле гораздо хуже – программисты ушли программировать и кто знает, что произойдет, когда они вернутся демонстрировать результат?

На одном из машиностроительных предприятий мы внедряли систему, автоматизирующую большинство основных бизнес-процессов, в том числе управление закупками, продажами, производством (включая производственное планирование) и др. Объем настроек был большой, стадия их реализации занимала почти три месяца. В то время (более пяти лет назад) у нас был не слишком большой опыт выполнения масштабных проектов. Соответственно, руководитель проекта не подумал о том, что такой длительный интервал работ надо бы разбить на более мелкие этапы. Программистам раздали задачи, приступили к выполнению.

Надо отметить, что даже если существенная часть работ по настройкам выполняется сотрудниками, которые находятся непосредственно на территории предприятия – в любом случае держать «руку на пульсе» настройки предприятия проектной команде сложнее, чем на начальных стадиях. Интенсивность взаимодействия с представителями подразделений неизбежно снижается, по сравнению с работами по проектированию.

На нашем проекте реализовались почти все из возможных потенциальных проблем. Предприятие вошло в холдинг, начали пересматривать целесообразность финансирования проекта. Длительный этап стадии настроек автоматически вел к высокой стоимости этапа, пришлось понервничать из-за рисков неполучения этих денег; кроме того, пришлось дополнительно обосновывать, почему стоимость такая высокая. Менеджеры отделов, для потребностей которых выполнялись настройки, имели представление о том, какой полезной работой занимается проектная команда. Но интенсивных контактов с программистами они, разумеется, не имели (в течение всех этих трех месяцев). Соответственно, кредит доверия к проекту с их стороны стал уже не так высок, как в начале, когда при моделировании и постановке задачи с ними общались почти ежедневно. И они уже не могли так уверенно и авторитетно подтвердить новому собственнику, что все три месяца проектная команда интенсивно занималась архиважнейшим делом:

- Ну да, вроде какие-то работы они сейчас выполняют. Какие именно и насколько это важно для нас? Откуда же нам знать! Что-то там программируют потихоньку, уже третий месяц...

И наконец, за прошедшее время поменялся приоритет в оптимизации управления – детализация данных о затратах на производственный заказ стала многократно важнее, чем задача внутрицехового планирования. На ее освоение решили не отвлекать начальников цехов, соответственно выполненные в этой части настройки системы оказались вообще не востребованными. А на них, между прочим, приходилось почти четверть всех трудозатрат – не так уж мало...

Проблема 9. Недостаточный оперативный контроль

«Отсутствие контроля? Что вы, на наших проектах такого не бывает!» Может показаться невероятным, но так действительно бывает. Руководитель проекта успешно определил цели, спланировал задачи, передал их непосредственным исполнителям и спокойно занялся чем-то другим. В том или ином виде слабый контроль легко наблюдать на очень многих проектах автоматизации. И наиболее показательны крайние случаи.

Вот что на полном серьезе рассказывал мне директор партнерской фирмы пару лет назад: - Интересный и сложный проект, на фабрике мороженого. Настраиваем подсистему учета затрат и расчета себестоимости. Ключевая задача поручена опытному программисту Коле Весельчакову (*имя изменено*), хорошо проявившему себя на предыдущих проектах. Работы по кодированию по планам примерно на два месяца. ТЗ написано достаточно конкретно, Коля приступил к программированию. И что ты думаешь? Через два месяца, когда надо сдавать работы заказчику, выясняется, что реализована в лучшем случае четвертая часть от всех требований! Я был в шоке, руководитель проекта тоже, про заказчика и говорить нечего. Но ничего, обошлось: подключили еще программистов, худо-бедно сдали работы – правда, с провальной рентабельностью и безнадежно сорванными сроками...

- Как же так? – спрашиваю я, - Его работу два месяца вообще никто не контролировал, что ли?
- Отчего ж? – отвечает коллега-директор, - за этот период мы с руководителем проекта не один, и не два (а целых три!) раза спрашивали Колю о состоянии дел. Весельчаков докладывал, что осталось чуть-чуть, еще несколько дней и будем сдавать подсистему досрочно. Не лезть же к нему программный код смотреть, правда?
- Почему бы и нет?
- Гм! - ответил директор. – Но раньше на этого программиста особых нареканий не было...

Как вы думаете, что случилось с Колей Весельчаковым, почему он говорил об одних результатах, а на деле выдал другие? Возможно, он искренне заблуждался в оценках прогресса выполнения задачи. А может быть, лгал намеренно. Возможно, сначала Коля взял хороший темп, но какое-то неординарное событие полностью лишило его работоспособности (любимая канарейка погибла в когтях не менее любимой домашней кошки?..). Или что-то еще. Это все не важно и не должно нас с вами интересовать. Важнее другое – простейшими элементами контроля вполне можно снизить влияние человеческого фактора на результат. И, что не менее важно: обеспечить прогнозируемость, то есть понимание – а все ли идет как планировалось и не нужно ли срочно предпринять корректирующие действия?

Конечно, пример моего коллеги экзотичен, такие запущенные случаи сейчас редкость. Но преимущество технологии как раз в том, что следуя ей, мы получим предсказуемый результат, независимо от персоналий. Например, в своей работе мы используем еженедельный Отчет о состоянии проекта. В этом документе по всем задачам отмечается: что планировалось сделать за неделю, что сделано на самом деле, какие проблемы и риски выявлены, что планируем дальше (формы и примеры таких Отчетов приведены в нашем дистанционном тренинге). И каждую неделю – демонстрация этих достигнутых результатов. Очевидный и простой способ контроля – но это действует.

*Можно использовать совсем другие методы, это не имеет никакого значения – лишь бы они были стандартизованы, **регулярны** и применялись на всех проектах.*

Еженедельный контроль должен быть привычкой фирмы: раз в неделю бывают выходные, раз в неделю бывает отчет и демонстрация результата (случается, что на какой-то неделе нет выходных или случается, что нет отчета – но это редкие, осознанные и обоснованные исключения). Если у нас по какому-то проекту нет отчетов неделю, две, три – мы понимаем, что растет риск получить «проект Весельчакова», кто бы ни работал на этом проекте. В зависимости от РП и исполнителей риск может быть больше или меньше – и мы стараемся его снижать до минимума, нормализовывая ситуацию с отчетами, просматривая результаты.

Только не надо проявлять чрезмерный фанатизм. Есть примеры, когда менеджеры изматывают участников проекта бесконечными отчетами и проверками по несколько раз в день. В результате проект попадает в так называемый «порочный круг контроля»: излишние проверки воспринимаются сотрудниками как недоверие и отвлечение от нормальной работы, это демотивирует исполнителей, что закономерно ведет к снижению работоспособности. Разумеется, следствием являются плохие результаты работ. Наблюдая ухудшение состояния проекта, менеджер спасается в усилении контроля, что еще больше демотивирует сотрудников и т.д.

Проблема 10. Растущие «междуэтапные» интервалы на сдачу-приемку

В начале каждого проекта обычно проходит установочное совещание с высшим руководством предприятия – чтобы все участники составили себе мнение, насколько заказчику важна автоматизация. На разных предприятиях приоритеты могут обозначать по-разному. Но зато почти каждый топ-менеджер на каждом из проектов декларирует, что они со своей стороны затягивать проект не будут и решения обещают принимать быстро.

К сожалению, на практике «быстро» получается редко... Одно из сильно тормозящих проект явлений – растущие интервалы приемки работ ответственными менеджерами. Надо заметить, ситуация типична почти для каждого проекта автоматизации, за небольшими исключениями.

Разумеется, сроки сдачи-приемки всегда жестко регламентированы (в том же Договоре, в Уставе проекта, можно и еще где-либо зафиксировать).

Обычно такой срок составляет пять (реже – десять) рабочих дней.

Но несмотря на регламентированность положения, от этого срока возникают отклонения. Почему и что с этим делать?

Базовая причина – менеджерам, конечно же, не хватает времени на то, чтобы отвлекаться от своей основной работы. Это нормально и это просто надо принять как факт.

Например, мы как-то автоматизировали предприятие в Прибалтике, на котором было по-европейски динамичное управление. И на котором директор был в высшей степени заинтересован в результате, прилагая для его достижения реальные усилия. Однако даже на таком «почти идеальном» проекте зачастую бывало сложно несколько дней подряд интенсивно третировать начальников отдела работами по приемке очередной стадии.

Что уж говорить про менее продвинутые производства с неторопливым менеджментом...

С этой составляющей проблемы сложно что-либо делать. Надо просто многократно (и до начала проекта, и в процессе) предупреждать **всех** ответственных со стороны предприятия: «Готовьтесь выделять на проект свое время! А особенно плотно – на приемку результата».

Другая причина затянутой приемки – опасения принимающих результат. Опасения относительно того, что результат будет не безупречен, а они его примут в таком безупречном виде. И что тогда? Менеджер может предполагать, что либо ему втюхают программу с ошибками и ему в дальнейшем так и придется на ней работать и мучиться. Либо, поскольку он эти ошибки не заметил – за трудозатраты на исправление ошибок придется отвечать ему.

С этими страхами вполне можно (и нужно) работать. На начальных стадиях, таких как функциональное моделирование – необходимо вовлекать пользователей в отработку примеров на типовом решении.

Об этом мы здесь писали в разделах выше...

И пояснять, что **концептуально** они сами убедились, что модель работоспособна. А если какие-то **не концептуальные** мелочи в модели упущены – реализовать их в дальнейшем не составит проблем.

На заключительных стадиях, когда принимается не модель, а настроенная система – надо обращать внимание на следующее:

- Во-первых, как бы тщательно не проверяли функционирование – в любом случае непременно будут появляться новые требования к системе (или объективно со стороны бизнеса, или субъективно – со стороны удобства работы пользователей). И так или иначе эти требования без проблем можно будет реализовать – можно в рамках сопровождения, а можно собственными силами предприятия, как пожелает руководство.
- Во-вторых, есть скрытые недостатки, которые вообще никак нельзя выявить до промышленной эксплуатации. Они могут проявиться в реальной работе через месяцы, при особой комбинации ситуаций. Это нормально и устранение таких скрытых недостатков у нас предусмотрено в договоре – по гарантии, нашими силами за наш счет.

А если ничего не помогает и приемка работ движется медленнее, чем сонно зевающая улитка? Это плохо, потому что затягивать проект никому не интересно (ни исполнителю, ни руководству предприятия – даже если оно бессильно что-либо с этим затягиванием сделать). Действуем по ситуации, вот пара примеров из практики.

Недавний проект – предприятие обрабатывает растительное сырье, управление весьма бюрократизированное, поэтому оперативность их работы не высока. В договоре предусмотрено пять дней на приемку. По прошествии четвертого дня заказчик извещает нас, что он никак не уложиться в этот срок, просит дать еще

пять дней. Мы понимаем, что с их бюрократизацией настаивать или искать компромиссы будет малорезультативным. Поэтому пишем официальное письмо: ладно, первый прецедент мы принимаем, но со своей стороны обозначаем, что растущие сроки – это растущие трудозатраты, это изменение концепции проекта. И при повторении прецедента обязательно нужно будет пересматривать полностью все плановые сроки и бюджеты. Помогло, следующие стадии принимали быстрее.

Другой пример, предприятие более склонно к переговорам и поиску совместных решений проблем. Заказчик еще при согласовании договора сразу обратил внимание, что они считают свои производственные процессы настолько сложными (химическое производство), что будут анализировать функциональную модель очень тщательно. И пять дней им никак не хватит, надо сразу предусмотреть пятнадцать. Да, можно и так, конечно – но это другие сроки и стоимость. В итоге договорились, что параллельно с приемкой проектной документации мы будем выполнять отдельные фрагменты работ из последующих стадий: подготовку нормативно-справочной информации, согласование входящих остатков. То есть будем делать такие элементы работ, которые не завязаны на результаты предыдущих стадий – их не очень много, но они есть. Это для всех оказалось приемлемо.

Проблема 11. Недостаточное внимание к управлению рисками

Управление рисками – это целая специально выделенная область стандарта управления проектами РМВоК. То есть, стандартом обозначено, что каждый руководитель проекта кроме задач из прочих девяти областей знаний (управление предметной областью, бюджетом, коммуникациями и др.) обязательно должен уделять внимание управлению рисками.

Это в теории. А на практике? Про то, что управлять рисками нужно, знает почти каждый руководитель проекта. Более того, некоторые опытные руководители проектов представляют себе, какие типовые риски могут приключиться. И еще некоторые из них знают о том, какие стратегии работы с рисками в каких случаях целесообразно применять. Но что интересно – в «реальной проектной жизни» рисками управляют буквально единицы!

Вполне понятно, почему это так. У руководителя проекта ворох реальных проблем, которые надо срочно решать. И когда ему, скажите, найти время на то, чтобы предотвращать потенциальные проблемы, которые еще не случились?

На самом деле, не секрет, что если-таки найти время на предотвращение проблем – то профилактические меры обычно оказываются менее трудозатратными, чем в дальнейшем

«гасить пожар», прилагая для этого немыслимые усилия! Хорошо, если вообще удастся эти уже возникшие проблемы успешно решить...

Поэтому мы стараемся придерживаться следующей практики. На подготовительной стадии руководитель проекта просматривает специальный реестр типовых рисков и оценивает, какие из них могут случиться на текущем конкретном проекте. Далее в Уставе проекта он отмечает эти риски и предлагает стратегию работы с ними. Похожий процесс повторяется перед началом каждой стадии. И в дополнение ко всему этому, еженедельно при составлении оперативного отчета о состоянии проекта Руководитель проекта фиксирует в этом отчете – не возникли ли угрозы еще каких-либо потенциальных проблем. И если возникли – что он собирается предпринимать для того, чтобы их контролировать.

А предпринимать в общем случае можно одну из следующих стратегий:

- избегание риска
- снижение риска
- защита от риска
- перемещение риска.

Надо признать, что тема работы с рисками емкая и на этот счет есть множество материалов. Это и отдельные монографии, и методические подборки, издаваемые фирмой 1С, и многочисленные статьи... Несколько более подробно практические примеры различных стратегий управления рисками, а также перечень типовых рисков (потенциальные «грабли», на которые легко наступить), рассмотрены в специальном бонусном разделе нашего дистанционного тренинга. Здесь же мы постарались, как минимум, привлечь внимание к вопросу, который многими вообще игнорируется.

Проблема 12. Проблемы при выборе типового решения

В чем, собственно, заключается проблема, когда предприятие выбирает типовое решение, которое будет являться основой для проекта автоматизации? Проблема в том, что сложно определить критерии, по которым выбирается система. Нужны простые методы оценки типовых решений, так как опыт и интуиция есть не у всех. Множество параметров готовых базовых продуктов не позволяют определить, какой из этих продуктов подходит наилучшим образом.

Коробочный продукт – это только «исходная точка», задающая вектор развития системы. Где гарантия, что из выбранного коробочного продукта именно в вашем случае удастся с приемлемым результатом построить систему, которая оправдывает понесенные на нее расходы?

Программа – необычайно податливый материал. Можно взять типовое решение и

переработать его в процессе адаптации к предприятию на 100%. Например, благодаря открытости платформы IC – все функции типового решения на внедрении можно переписать заново, если потребуется.

При выборе потребители рассматривают типовые решения на предмет минимальных доработок – это критерий. Но не всегда принимается во внимание, каков характер требуемых доработок!

При попытке проанализировать некое типовое решение на предмет «предусмотрено/не предусмотрено» потребитель сталкивается с огромным количеством функций, форм, справочников, документов и т.д.

Мы часто видим, как потребители строят сравнительную таблицу функционала разных решений и после этого приступают к медитации над этой таблицей.

Пытаться проанализировать все это хозяйство «в лоб» - довольно затруднительно и не каждому по силам. Как быть? Есть простой алгоритм.

Надо ранжировать все возможности типового решения на два уровня:

- Уровень функциональности ядра (возможность системы в принципе хранить и рассчитывать необходимые данные)
- Сервисные функции (например, удобные формы пользовательских отчетов, автоматизированные подборки данных в документы, подсказки пользователю и т.д.)

Конечно, программа с хорошо реализованным взаимодействием между пользователем и компьютером – очень привлекательна. Вот только в крупных корпоративных системах эти показатели отходят на второй план. Повышение комфортности работы «простых» пользователей – не цель дорогостоящего внедрения, не так ли?

Причем перечень сервисных параметров очень обширный и анализировать его непросто...

А вот требования к базовой функциональности (какие данные система способна хранить и обрабатывать) – более универсальны и более однозначны.

Представим себе, что менеджеру закупок нужно создать автоматически заявки поставщикам на основании рассчитанной потребности в материалах. Простая формулировка функции таит за собой огромное количество вариантов ее воплощения и особенностей бизнес-процесса. Это функциональность взаимодействия пользователя с компьютером. В типовом решении может быть заложено только ограниченное количество моделей (или вообще только одна), и совсем не очевидно, что эти модели подойдут полностью без доработок. С другой стороны, такое базовое требование как «Система должна хранить рассчитанную потребность в материалах на даты количественном выражении в разрезе заказов клиентов» является однозначным, и реализация этой функции в конкретном решении достаточно легко поддается анализу. Например, в какой-то

системе может оказаться, что в разрезе заказов получить потребность в принципе нельзя – она рассчитывается как сводная по всем заказам. Это ни что иное, как ограничение системы (она не способна рассчитывать и хранить эти данные), которое придется преодолевать путем доработок.

Мы предлагаем на этапе выбора типового решения основное внимание уделить наиболее важному уровню – ядру системы. На этом уровне перечень анализируемых параметров резко сокращается, анализируемые параметры более конкретны и понятны.

Вот один из примеров с проекта наших партнеров. Объект автоматизации – крохотная фирма, собирающая аналогово-цифровые приборы, производство сводится фактически только к операциям сборки комплектов. На первом этапе наиболее критичным вопросом выделили автоматизацию складских операций. Менеджеры предприятия понимают, что в дальнейшем их потребности в информации для управления наверняка вырастут. Тем не менее коробочный продукт выбирают исходя, в первую очередь, из удобства для пользователей. А на втором плане – стоимость коробки. В результате выбрали типовое решение от 1С, ориентированное в основном на бизнес-процессы торговли (но простейшие функции производства там отразить вполне можно).

Для текущих задач продукт полностью подходит, а если что – потом можно доработать собственными силами.

Проект успешно завершен, все довольны. Через время – экономическая ситуация осложняется, руководство уделяет больше внимания производственному планированию, некоторые изменения происходят в производственном процессе. В производстве тоже изменения – часть комплектующих-полуфабрикатов начали делать сами, процесс становится многопередельным. Появляется задача расчета потребности в комплектующих. Хотели доработать программу, но собственными силами не справились, в итоге наткнулись на прямо-таки астрономическую стоимость доработок. Фактически, доработки означали создание в системе функционала полноценного производственного планирования и учета. Который, заметим, есть в готовом виде в комплексном типовом решении 1С, но, понятное дело, отсутствует в торговой программе. После мучительных раздумий и попыток самим написать «на коленках» ERP-систему – предприятие перешло на другой, комплексный продукт, понеся существенные дополнительные затраты.

Вывод простой – все требования к АС должны быть поделены (ранжированы) на требования к ядру и требования к сервисам взаимодействия (настройка над ядром), и выбор решения должен быть в первую очередь основан на соответствии функциональности ядра требованиям, предъявляемым к ядру. С расчетом на то, что доработки сервисов взаимодействия обладают несущественными рисками и могут быть легко выполнены под специфику предприятия.

Проблема 13. Полное непонимание политической ситуации

Интересен пример, как мой коллега, на тот момент уже обладающий большим техническим опытом, попал на проект с нетривиальной политической ситуацией. До этого он выполнял несколько технически сложных проектов, но на них отсутствовали конфликты заинтересованных сторон. И поэтому область политики была для него в новинку.

Предприятие представляло собой крупнейшее машиностроительное производство, расположенное в небольшом провинциальном городке. Новый собственник завода посчитал необходимым повысить прозрачность учета средствами автоматизации. Фактически это означало, что ему потребовалось получать учетные данные хотя бы (для начала) о состоянии запасов, производства и взаиморасчетов с контрагентами – на момент приобретения им завода никто не мог предоставить никакой более-менее достоверной информации, за исключением регламентированной бухгалтерской отчетности.

Итак, на предприятии множество влиятельных заинтересованных сторон. Собственник заинтересован в успешности проекта. Директора завода не слишком заботит автоматизация, но он поддерживает проект, потому что этого требует собственник. Отдел снабжения, который контролирует запасы на складах – не против системы, но им вечно некогда.

Ну а как же – ведь если на минутку отвлечься от работы с поставщиками, то сразу начнутся перебои в снабжении материалами и комплектующими. Нельзя же допустить, чтобы производство остановилось из-за какой там автоматизации, верно?

Финансовый директор пришел недавно, из команды собственника. Поэтому для него главное – продемонстрировать свою состоятельность как менеджера. Если проект этому поспособствует – тогда ладно, хорошо. А если нет – проектную команду лучше аккуратно выдворить с завода, выставив их некомпетентными.

Есть еще Отдел АСУ (ИТ) - с одной стороны, они хотят быть причастными к успешному проекту. С другой стороны, они боятся потерять свою значимость, когда все их самописные программы будут снесены и замещены новым технологичным решением.

Начальник направления запчастей не допустит успеха проекта ни в коем случае!

Потому что запчасти всегда востребованы населением и сколько запчастей потихоньку вышло своим ходом за проходную без всяких накладных – собственнику знать об этом совсем не нужно.

Будущие пользователи системы, рядовые сотрудники завода, которые входят в проектную команду и выполняют существенный объем работ (выверяют нормативно-справочную информацию, вводят входящие остатки и т.д.) – больше всего они хотели бы, чтобы их оставили в покое. С их слов, они на свою ежемесячную зарплату обеспечивать минимальный жизненный уровень никак не могут, при всем желании. Поэтому у них у

каждого огород и надо уже начинать картошку сажать. А то зимой есть будет нечего. В сверхурочной надбавке за участие в проекте автоматизации финансовый директор отказал. Какие уж тут могут быть сверхурочные работы...

Главный инженер (второе лицо на заводе!) собирается на пенсию. От его слова зависит многое. Он благодушен ко всем... и к нам тоже... но он нетороплив. И ни на какой конфликт ни за что не пойдет – ему это уже незначит.

Как видим, у всех свои интересы. У кого-то интересы могут способствовать проекту автоматизации, у кого-то наоборот... Но ни у одной заинтересованной стороны, кроме собственника, проект не входит в непосредственные целевые приоритеты.

*И вот в этом клубке политических интриг завяз наш неискушенный
Руководитель проекта.*

Что нужно было делать в такой ситуации? Как скалолаз на сложном подъеме делает запланированные остановки, так и нам нужно было обязательно планировать и демонстрировать **промежуточные положительные результаты**. Руководитель проекта в сложной политической обстановке должен «делать зарубки», закрепляться на достигнутых уровнях, обеспечивая очевидность успешности уже пройденного пути. Это является термоядерным катализатором для усиления воздействия лиц, поддерживающих проект. И одновременно, это снижает результативность нападок противников.

Что НЕ нужно было делать? Ни под каким видом нельзя было проявлять признаки политической слабости. Когда пользователи не успевали выполнить поставленные задачи, когда направление запчастей саботировало проект, когда снабженцы не могли выделить время – по каждому чиху мы сразу просили помощи у высшего руководства предприятия. Конечно, административный ресурс нужен, но его надо применять дозированно. При переходе некой тонкой грани перебор административного ресурса трактуется противниками проекта как недостаток у проектной команды собственной воли. Это сразу же делает проект уязвимым для атаки с их стороны.

Теоретические концепции рекомендуют ряд действий для сложных проектов, в том числе: понять заинтересованные стороны и степень их влияние на проект, управлять ожиданиями заинтересованных сторон и т.д. Это в равной степени и безупречно верно, и банально-академично. Руководитель проекта должен выполнять эти действия, но если он не политик, то он не сможет их выполнять. То есть на проекты со сложной политикой должен назначаться **лидер с опытом работы** в таких условиях.

Проблема 14. Иллюзия о невозможности остановить проект

Крайне редко, но бывает и такое, что проект заходит в полностью тупиковую ситуацию. Концепции управления проектами допускают и идентифицируют *прекращение проекта* как одно из возможных действий при реализации рисков (или эскалации проблемы). Иногда это становится единственным адекватным решением, однако на практике почти что каждый руководитель проекта обременен иллюзией, что проект прекращать нельзя ни при каких условиях. Конечно, речь не идет о том, чтобы бросить все на полдороге и уйти. Требуется проявление воли руководителей высшего звена, с тем чтобы проделанная работа не пропала зря, а достигнутые результаты были зафиксированы (или в худшем случае – выйти из проекта с минимальными убытками для всех сторон). Поскольку если проект «сваливается в пике» и его вовремя не остановить – потери будут только нарастать.

В нашей истории был отмечен случай, когда проект приходилось прекращать. Это происходило достаточно давно, детали уже не сохранились, но суть все участники проекта хорошо помнят до сих пор.

Автоматизации подлежала одна из бизнес-единиц крупного строительного холдинга, реализовывались задачи управления закупками, запасами, финансовыми ресурсами и расчет себестоимости оказываемых услуг. Примерно через полгода в опытную эксплуатацию успешно были введены почти все участки. Однако в последнем функциональном блоке – расчете себестоимости – один из видов документов для оформления наряда на услуги приходилось заполнять вручную, в то время как по ТЗ было предусмотрено автоматическое заполнение сметных данных. Как выяснилось, обеспечить это автоматическое заполнение оказалось чрезвычайно сложной задачей – алгоритм зависел от мельчайших деталей, которые было трудно исчерпывающе формализовать, кроме того, руководитель проекта при реализации других функций допустил несколько некорректных технических решений, которые еще больше усложняли это автозаполнение.

Срок сдачи проекта в целом был намечен примерно по истечении полугода после начала работ. А сдать проект невозможно – руководство предприятия не спешит принимать систему в промышленную эксплуатацию и проводить окончательные расчеты с исполнителем, справедливо утверждая:

- Погодите, а где же автоматическое заполнение наряда? Да, все остальное работает. Да, себестоимость считается, все правильно. Но вот это ручное заполнение нам не нравится, так не договаривались!

Согласовали перенос сдачи работ на две недели. Руководитель проекта уверен, что за это время мы уточним детали алгоритма и нам удастся реализовать пресловутое автоматическое заполнение. Напомню, система в это время успешно функционирует в режиме опытной эксплуатации, причем по всем блокам, только в последнем – часть расчета приходится делать вручную. За три недели решить задачу не удалось.

Руководитель проекта запросил еще две недели – ок, согласовали, утвердили. Затем еще две недели... Через почти что три месяца (!) после запланированного срока сдачи руководитель проекта был уверен, что осталось чуть-чуть и решение неизбежно будет найдено. И его можно понять – осталась такая малость, какой-то единственный жалкий документ препятствует нормальному закрытию проекта, с соблюдением всех взаимных обязательств всех сторон. Доделать этот расчет для него уже стало делом чести.

Но, наконец, пришло время кураторам проекта трезво взглянуть на ситуацию: откладывать промышленную эксплуатацию еще дальше – явно противоречит интересам всех сторон. На высшем уровне была зафиксирована дата, после которой все работы в любом случае должны быть прекращены и система должна быть передана в промышленную эксплуатацию как есть.

Взаиморасчеты сторон при этом определялись отдельным соглашением, в зависимости от достигнутых результатов.

Конечно же, к этой окончательной дате автозаполнение так и не реализовали. Что, впрочем, не помешало предприятию успешно пользоваться системой, хоть и с оговоренными ограничениями. Позже руководитель проекта признался, что ему было сложно «вынырнуть» из затянувшегося этапа и если бы проект директивно не прекратили – он бы так и продолжал пытаться доделать все «как надо» любой ценой, месяц за месяцем...

Что делать со всеми этими проблемами?

Для начала полезно будет проанализировать выполняемый вами проект автоматизации. Может ли на нем реализоваться одна из перечисленных проблем? Не поленитесь, продумайте заранее методы работы с этими рисками. Если удастся избежать хотя бы одной из типовых ошибок, то это уже даст реальный эффект.

Который можно измерить в деньгах, сроках, трудовых затратах.

А как стартовать проекты правильно с самого начала? Для этого, к сожалению, недостаточно знаний. Нужен опыт, нужны **навыки**... Навыки вполне можно получить самостоятельно, для этого всего лишь требуется некоторое время.

Что делать тем, у кого нет лишнего времени? Как отмечено в начале этой книги, в нее мы включили всего лишь несколько зарисовок с реальных проектов. Однако формат книги не позволяет обеспечить достаточную интерактивность, чтобы полностью **встроить технологию в реальную работу**. Зато эту интерактивность мы воплотили в нашем дистанционном тренинге по проектным технологиям – в практических заданиях, роликах-кейсах, тестовых вопросах и т.д.

Ну и для особых случаев, когда ситуация стала тупиковой и перейти от набивания шишек к технологичной работе надо «уже вчера» - мы тоже можем помочь. Мы индивидуально проведем «за руку» по всем аспектам проектной технологии, с учетом всех особенностей именно вашей ситуации, **с гарантией** результата.

Пишите: info@itrp.ru

или info@planfact2.ru

Для заметок