### Требования к проекту

#### Введение

Относительно свободная форма изложения, принятая авторами, допускает введение новых терминов и соответствующих им понятий, так сказать, по ходу изложения материала, не обременяя читателя отдельно вынесенными определениями или описаниями. Но это только там, где авторы надеются на опыт и интуицию читателя. Например, с самого начала используется термин требования к проекту (или, коротко, требования), обозначающий "то, что под этим понимают компетентные люди".

Но есть термины, с которыми подобные вольности недопустимы. В статье вводится понятие область требований. В российских стандартах существует термин предметная область, а в украинских — предметна сфера, однако область требований обозначает нечто принципиально другое.

Почему именно "область", а не класс, к примеру, или группа? Тут две причины:

* Часто необходимо учитывать не только актуально существующие требования, но и требования, которых сейчас нет, но которые потенциально могут возникнуть. Кроме того, иногда необходимо принимать во внимание именно факт отсутствия требований, обладающих определенными свойствами. В этих случаях термин пустая область воспринимается гораздо естественнее, чем, скажем, пустой класс или пустая группа.
* Для визуализации предлагается авторский тип диаграмм, где каждой области требований соответствует графическая область на диаграмме.

Области требований могут частично перекрываться, равно как одна из них может полностью содержать другую, отражая тем самым какое-либо из свойств принадлежащих им требований. Области требований очень похожи на диаграммы Венна из теории множеств. И чем в большей степени каждое отдельное требование будет обладать свойствами атомарности и отличимости (обязательные свойства для элемента множества), тем больше математических результатов из теории множеств можно будет применить.

Предлагаемый метод построения диаграмм областей является инструментом аналитика для процедуры начальной классификации требований. В процессе выделения существенных в данном конкретном случае областей, определения их взаимосвязей относительно включения-пересечения и распределения имеющихся требований по соответствующим областям аналитик приходит ко все более точному и глубокому пониманию сути самих требований.

По ходу изложения материала диаграмма будет эволюционировать от вспомогательного и совсем не обязательного инструмента визуализации простых и очевидных понятий до самостоятельного, мощного механизма, несущего глубокую семантику и, при необходимости, являющегося прочной базой для дальнейшей формализации и применения математических методов.

#### Диаграмма областей требований

#### Простейший случай диаграмм

Итак, в простейшем случае на диаграмме имеем область всех мыслимых и "немыслимых" требований — наш аналог универсального множества в математике.

В рамках этого универсального множества всех требований выделяем две большие области: область компетенции и состоятельности Заказчика и область компетенции и состоятельности Разработчика.

Таким образом, есть Заказчик и Разработчик, у каждого из них — своя область компетенции и состоятельности. Чем дальше в сторону информационных технологий смещаемся, тем шире область компетенции и состоятельности Разработчика. Соответственно, чем дальше в противоположную сторону (сторону технологий из предметной области) — тем меньше специалисты разработчика показывают сообразительности и готовых наработок.

С другой стороны, чем более мощные информационные технологии предлагаются Заказчику, тем меньше у него находится соответствующих задач и тем меньше сотрудников, способных разобраться как в самих задачах, так и в предлагаемых методах автоматизации их решения.

Диалог между представителями Заказчика и Разработчика становится конструктивным только тогда, когда области их компетентности и состоятельности начинают пересекаться.

Другими словами, информационное образование экспертов Заказчика должно позволять им формулировать свои требования пусть на ломаном, но все-таки "человеческом" языке, с одной стороны. А с другой — иметь элементарные представления об автоматизации для необходимой оценки и критики предлагаемых Разработчиком методов.

В свою очередь, опыт аналитиков Разработчика должен достичь такого уровня, чтобы они имели возможность разобраться в плохо структурированных лекциях экспертов Заказчика, состоящих на 80% из изложения частных случаев и исключений, каждый из которых просто "вопиит гласом велиим" о высочайшем уровне и исключительности данного эксперта в его области деятельности.

Но серьезное сотрудничество начинается только тогда, когда на пересечении этих двух областей, которое можно назвать областью взаимопонимания, накапливается некая "критическая масса" требований. Именно эти требования и являются основным материалом для переговоров и обоснования целесообразности самых ранних этапов разработки проекта.

Замечательно, что этими требованиями, как правило, не может пренебречь даже высшее руководство — как с одной, так и с другой стороны. Как правило, эти требования в обязательном порядке попадают в официальные документы и именно на них "высокие договаривающиеся стороны" своими широкими руководящими жестами ставят две черты: порог рациональной автоматизации и порог рациональной формализации предметной области Заказчика.

Эти две руководящие директивные черты отрезают две небольшие, но очень интересные области:

* излишние требования к автоматизации от особо грамотных сотрудников Заказчика;
* излишняя формализация предметной области от особо рьяных аналитиков Разработчика.

Примером требования из первой области может служить пожелание системного администратора Заказчика: получить систему автоматического архивирования базы данных корпорации на какое-нибудь нестандартное устройство со своим специфическим программным обеспечением по его управлению (какой-нибудь древний стример или новейший CD-RW, оборудованный механическим чейнджером).

Пример для второй области — система автоматического ценообразования для работы дизайнеров или парикмахеров, основанная на эстетических признаках.

Как всегда, сроки — сжаты, бюджет — ограничен. А потому от руководства дается однозначная установка: никаких новшеств, никаких изысканий, реализуем только четко очерченную область требований — просчитать сроки и бюджет.

Итак, сроки и бюджет скрупулезно подсчитаны, скорректированы и утверждены на высшем уровне.

Приступаем к воплощению… и с ужасом обнаруживаем: для реализации требования В необходима реализация требований, оставшихся за пределами внимания руководства,— А или C.

В экстренном порядке собирается совещание, на котором руководство Разработчика (наконец-то) хоть немного прислушивается к мнению непосредственных исполнителей и на котором за бурными дебатами прячется основная цель: решить, кто же будет расплачиваться за промах. Руководство ли, вынужденное оплатить реализацию одного из требований A или C, — или же непосредственные исполнители, вынужденные реализовать все те же требования A или C, но (!) без дополнительных финансовых и человеческих ресурсов (в лучшем случае будет дана отсрочка по времени, потому что, как правило, все делается за счет "авралов" и "переработок").

Итак, приходим к пониманию, что существует еще одна область (рис. 7). Область требований, удовлетворение которых обязательно для реализации проекта в целом и, в частности, для реализации задокументированных и утвержденных требований. Ограничение ресурсов на реализацию этих требований или попытка обойтись вообще без их реализации ведет прямиком к завалу проекта.

Как только руководство принимает решение — требования из этой области не реализовывать — исполнители встают перед необходимостью делать это за свой счет. В результате наблюдается явление откачки средств и ресурсов из среды исполнителей. Это может выразиться в переработках, задействовании личных наработок или скрытого аутсорсинга исполнителей.

В любом случае результатом будут напряженные отношения между высшим руководством и исполнителями, грозящие перерасти сначала в скрытый, а потом и в явный конфликт. В случае если трудовые отношения построены по принципу "от конечного результата", пострадавшей стороной (то есть стороной, несущей потери) окажется непосредственный исполнитель. В случае же отношений с повременной оплатой все финансовые потери ложатся на плечи руководства Разработчика.

И в том, и в другом случае Заказчику извне ситуация с развитием проекта кажется неопределенной, но, тем не менее, не предвещающей больших тревог — хотя именно в это время закладывается мина замедленного действия под успех проекта в частности и сотрудничество вообще. Тогда как в случае проявления инициативы со стороны Заказчика в сторону переговоров с целью увеличения сметы проекта катастрофические процессы могут быть погашены.

Проще говоря, рассчитывать на конструктивную деятельность со стороны своего руководства исполнителям Разработчика не приходится. Провести переговоры и добиться разумного и обоснованного увеличения бюджета — дело немыслимое, а проявление инициативы в этом направлении со стороны руководства Разработчика вообще принято считать дурным тоном.

С другой стороны, предложение со стороны Заказчика — увеличить бюджет проекта — помимо того что дает возможность решить проблему, еще и работает в сторону повышения имиджа не только Заказчика, но и Разработчика.

Однако на практике мы нередко наблюдаем весьма печальную и уродливую картину. Руководство Разработчика, зажатое реальными и весьма серьезными проблемами по реализации упущенных требований с одной стороны и мнимыми, но весьма болезненными проблемами по преодолению имиджевого стереотипа поведения с другой, все-таки вынуждено действовать. События развиваются по плану, который, может, и кажется счастливой находкой — но на деле лишь оттягивает момент кризиса, приводя к еще более тяжелой ситуации.

На переговорах Руководство разработчика подает сложившуюся ситуацию в очень выгодном (но только при поверхностном взгляде) свете: во время работы над проектом оказалось, что в общем ключе работ, не изменяя идеологии, возможно получить намного более привлекательный для Заказчика продукт, чем предполагалось изначально. Для этого необходимо всего лишь незначительно увеличить бюджет, но (!) прямо на текущей стадии разработки проекта.

Очевидными плодами такого подхода будут:

* получение дополнительного финансирования, едва достаточного для закрытия образовавшегося прорыва и относительно благополучного завершения проекта в его изначальной редакции;
* обнародование необходимых, но упущенных пунктов, под видом доработок, составляющих новизну и привлекательность проекта в его новой редакции;
* целый пакет новых и весьма амбициозных требований, главными свойствами которых будут повышенная туманность формулировок и, как следствие, повышенные риски в реализации;
* фактические затраты на реализацию этих новых требований окажутся, в лучшем случае, большими не на порядок, а лишь в несколько раз.

Есть простой признак, позволяющий Заказчику распознать такую ситуацию. Предложения об улучшениях поступают от Разработчика в конце временной цепочки его состояний: спад активности — неопределенность — бурная внутренняя деятельность — предложение чего-то лучшего. Вот уж действительно, лучшее — враг хорошего.

Напротив, если предложения об улучшении явились результатом ровной работы аналитиков Разработчика, да еще и при болезненном для Заказчика вскрытии его внутренних проблем,— тогда это может служить поводом к доверительной оценке таких предложений.

Существует еще одна область требований, реализация которых необходима для поддержания естественной и полноценной работы системы в соответствии со взаимно понятными и задокументированными требованиями (рис. 8). Изначально, требования из этой области не были замечены аналитиками Разработчика и считались в равной мере как очевидными, так и обязательными, с точки зрения экспертов Заказчика.

Если проигнорировать реализацию этих Требований, то Заказчик получит Систему, в которой функционирование автоматической части должно поддерживаться большим объемом окружающего ручного труда. Грубо говоря, придется заводить штат сотрудников, специалистов Предметной Области, работающих на Систему.

Авторам пришлось внедрять сторонний продукт, в котором был реализован автоматический выбор варианта делопроизводства в зависимости от текущего состояния счета клиента. При этом велся строгий учет самих платежных документов и даты их прихода. Главным упущением аналитиков Разработчика оказалась реализация системы как автомата, реагирующего на наличие определенного платежного документа, а не на общий баланс по счету. Надо было видеть мучения делопроизводителей, вынужденных в угоду системе редактировать пересчитанные вручную суммы по соответствующим документам, а то и вводить дополнительные фиктивные документы…

Да простит меня читатель за случившееся отступление от темы статьи: формальная модель для классификации требований. Прорвались наболевшие наработки по родственному, но все-таки другому направлению — "Как гарантированно завалить проект" (конструктивное название должно быть таким "Риски ошибок в руководстве проектом: какими они бывают и как их избежать"). Возможно, это действительно станет названием следующей статьи, пока же приведенные язвительные примеры послужат лишь подтверждением тому, что за видимой сухостью и формальностью предлагаемой диаграммы кроются болезненные вопросы, сплошь и рядом возникающие в реальных, "живых" проектах. А для обсуждения подобных наболевших вопросов ой как нужна хорошая их визуализация!

Но вернемся к нашей диаграмме и выделенной там области… и зададимся вопросом: "А почему, собственно, область поддерживающих требований должна иметь именно такой вид?" Почему не так, как на рис. 9?

Другой интересный вопрос: а какой смысл в двух различных подобластях данной области? Чем, по сути, отличаются требования A и C? Но об этом чуть далее.

А по перовому вопросу очевидно, что диаграмма на приведенном рисунке является просто проявлением более глубокого подхода и дает более интересные результаты, обсуждение которых следовало бы вынести за пределы данной ознакомительной статьи. Мы же ограничимся упрощенным вариантом (рис. 7) и пойдем далее.

Итак, имеем две зеркальные — как по расположению на диаграмме, так и по своим проявлениям в процессе разработки,— области:

* область контекста реализации — область поддерживающих требований, необходимых для полноценной реализации Разработчиком задокументированной части проекта;
* область контекста использования — зеркальная предыдущей и включающая требования, которые остались за пределами документов, но обязательны для полноценной работы заказчика.

Теперь настало время "показать семантику второго измерения". Проще говоря, ответить на поставленный ранее вопрос: а чем, собственно, отличается левая подобласть от симметричной ей правой? Если для полноценной работы Заказчика в соответствии с требованием E необходима реализация одного из поддерживающих требований D или F (рис. 9), то чем они между собой отличаются и что общего у них с зеркальными им требованиями A и C?

Внимательный читатель уже заметил две стрелочки в верхней и нижней части рисунка, подписи которых имеют простой смысл: чем левее на рисунке расположена точка, обозначающая требования, тем больше это требование опирается на широко известные и общепринятые методы решения, стандартное и распространенное оборудование, соответствующее программное обеспечение и т.д. А чем правее — тем больше особенностей в методах решения (вплоть до ноу-хау и патентов), тем больше специального оборудования и\или специального ПО.

И не случайно стрелочка, указывающая стремление использовать стандартные методы, расположена в верхней части, соответствующей стремлениям Разработчика. Именно от него зачастую звучат размашистые предложения в стиле: да, для решения задач такой серьезной и представительной организации, как Заказчик, ну просто необходимы: СУБД — не меньше и не дешевле Oracle 9, CRM-системы — не ниже самого "Сибел-системз", а для решения задач коммуникации без приобретения спутника с антеннами вообще не обойтись.

В ответ на такие предложения со стороны Заказчика активно выдвигаются возражения, что подобная мощь (читай — такие большие расходы) ему абсолютно ни к чему, что вполне достаточно существующей базы и что сотрудники Заказчика успешно справляются со своими задачами чуть ли не с калькулятором и на бумаге …

Существует лишь несколько задач, реально нуждающихся в автоматизации. Именно они являются особенностью деятельности Заказчика, их решение должно дать Заказчику конкурентные преимущества и т.д.

Изюминкой оказывается комплекс из уникальности самих задач, методов их решения и бессмысленности автоматизации всего остального без автоматизации именно этого…

В данном случае особенно к месту оказывается следующий пример из области делопроизводства. Существует обязательное требование E: система должна автоматически реагировать на состояние счета клиента и конфигурацию платежных документов. Существуют особые случаи, в которых необходим перерасчет сумм по платежным документам и реализация требования E нуждается в удовлетворении одного из требований D или F. Пример для требования F — наличие какого-то удобного и логичного механизма для перерасчета. Требование D — ужесточение (стандартизация) требований на платежные документы, исключающие возникновение запутанных ситуаций.

Для зеркальных требований A, B, C можно привести такой пример. B — набор отчетов, генерируемых из БД. Особенность — один или несколько из них не реализуются в виде единственного, пусть и сложного, SQL-запроса. Необходимы хранимые процедуры. Для реализации этого требования следует реализовать либо требование A — использовать Т-SQL с соответствующими механизмами MS-SQL Server (стандартное решение — и никакой головной боли Разработчику) либо требование С — использовать ODBC-интерфейс к файлу MS Access, на чем настаивает Заказчик. Но тогда Разработчику придется засучить рукава — и разрабатывать собственный эмулятор хранимых процедур. Вот вам и уникальность, и специфичность разработки.

Собственно, на этом ознакомительную статью можно и закончить. Заинтересованный читатель уже в состоянии самостоятельно развивать далее предложенный подход, что нас только порадует.