Требования к информационной системе и модели жизненного цикла

Развитие технологии разработки программного обеспечения, методов моделирования, появление CASE-технологий не решило проблему определения и формализации требований к информационным системам, но способствовало возникновению нескольких основных подходов. В статье рассматриваются проблема определения требований к информационной системе предприятия: выбора модели жизненного цикла (ЖЦ) разработки, определения контрактных условий, выбор нотации и инструментального средства формализованного описания требований.

Необходимость определения требований к информационной системе возникает в следующих случаях: в момент выбора новой информационной системы, при подготовке тендерной документации, при заключении договора на разработку или настройку выбранной информационной системы, при уточнении (детализации) потребностей бизнеса в процессе разработки или настройки системы, а так же при необходимости внесения изменений в систему в ходе эксплуатации.

В каждом случае перед специалистами предприятия и организации встает задача выбора уровня детализации требований, методов описания, включая формализованное описание с использованием графического моделирования. Какие факторы следует учесть, что бы выбрать оптимальный уровень детализации требований и наилучших метод их определения и формализации?

На уровень детализации, область определения, а так же используемые методы описания влияют:

- выбранная модель жизненного цикла разработки и внедрения;
- характера разрабатываемого и внедряемого ПО (заказная разработка, настройка информационной системы).
- используемые средств и методов проектирования (в случае заказной разработки).

В зависимости от этих параметров следует использовать ту или иную методологию моделирования, которая определяет выбор нотации, а выбор нотации, в свою очередь, определяет используемые инструментальные средства.

Модель жизненного цикла - структура, содержащая процессы, действия и задачи, которые осуществляются в ходе разработки, функционирования и сопровождения программного продукта в течение всей жизни системы, от определения требований до завершения ее использования. Существует несколько моделей и стандартов, в той или иной степени регламентирующих жизненный цикл, большинство из них относятся к заказному ПО (автоматизированным системам АС, и др.) и кроме непосредственно ЖЦ регламентируют также и процессы разработки:

- ГОСТ 34.601-90 распространяется на автоматизированные системы и устанавливает стадии и этапы их создания. Кроме того, в стандарте содержится описание содержания работ на каждом этапе. Стадии и этапы работы, закрепленные в стандарте, в большей степени соответствуют каскадной модели жизненного цикла [1].
- ISO/IEC 12207:1995 стандарт на процессы и организацию жизненного цикла. Распространяется на все виды заказного ПО. Стандарт не содержит описания фаз, стадий этапов.
- Custom Development Method (и, методика Oracle) по разработке прикладных информационных систем под заказ - конкретный материал, детализированный до уровня заготовок проектных документов, рассчитанных на использование в проектах с применением Oracle.
- Степень адаптивности CDM ограничивается тремя моделями ЖЦ: "классическая" (предусмотрены все работы/задачи и этапы), "быстрая разработка" (Fast Track), "облегченный подход", рекомендуемый в случае малых проектов и возможности быстро прототипировать приложения.
- Rational Unified Process (RUP) предлагает итеративную модель разработки,

включающую четыре фазы: начало, исследование, построение и внедрение. Каждая фаза может быть разбита на этапы (итерации), в результате которых выпускается версия для внутреннего или внешнего использования. Прохождение через четыре основные фазы называется циклом разработки, каждый цикл завершается генерацией версии системы. Если после этого работа над проектом не прекращается, то полученный продукт продолжает развиваться и снова минует те же фазы [2]. Суть работы в рамках RUP - это создание и сопровождение моделей, а не бумажных документов, поэтому этот процесс привязан к использованию конкретных средств моделирования (UML), а так же конкретной технологии проектирования и разработки (объектно-ориентированный анализ, object-oriented analysis, OOA, объектно-ориентированное программирование, object-oriented programming, OOP).

- Microsoft Solution Framework (MSF) сходна с RUP, так же включает четыре фазы: анализ, проектирование, разработка, стабилизация, является итерационной, предполагает использование объектно-ориентированного моделирования. [7]. MSF в сравнении с RUP в большей степени ориентирована на разработку бизнес-приложений.
- Extreme Programming (XP). Экстремальное программирование является самым новым среди рассматриваемых методологий, сформировалось в 1996 году. В основе методологии командная работа, эффективная коммуникация между заказчиком и исполнителем в течение всего проекта по разработке ИС, а разработка ведется с использованием последовательно дорабатываемых прототипов.

Сравнительный анализ моделей процессов Oracle CDM, ГОСТ 34, ISO/IEC 12207 приведен в [6]. Приведенный перечень далеко не полный, разработчики крупных информационных систем и компании-интеграторы так же предлагают свои методологии внедрения, содержащие основные этапы (модель жизненного цикла), формы документов, перечни вопросов и инструменты моделирования. Компания IBM внесла значительный вклад в теорию проектирования и разработки информационных систем, предложив еще в середине 1970-х годов методологию BSP (Business System Planning, система организационного планирования).

Все существующие сегодня методики определения требований к ИС являются наследниками BSP, используют предложенные в ней методы сбора информации, подходы в определении приоритетов требований, обеспечении полноты и непротиворечивости требований [4, стр. 75] . Структурирование информации с использованием матриц пересечения бизнес-процессов, функциональных подразделений, систем обработки данных (информационных систем), информационных объектов, документов и баз данных, предложенный в BSP, используется сегодня не только в ИТ проектах, но и проектах по реинжинирингу бизнес-процессов, изменению организационной структуры. Важнейшие шаги процесса BSP, их последовательность (получить поддержку высшего руководства, определить процессы предприятия, определить классы данных, провести интервью, обработать и организовать данные интервью) можно встретить практически во всех формальных методиках, а так же и в проектах, реализуемых на практике.

Различают две основные формальные модели ЖЦ: каскадную (последовательную) и спиральную (итерационную). В соответствии с каскадной моделью переход на следующий этап может происходить только после завершения предыдущего. Спиральная модель предполагает циклическое выполнение всех этапов каскадной модели, в результате чего реализуемость технических решений проверяется с помощью прототипов. Каждый виток спирали соответствует созданию фрагмента или версии ПО, на нем уточняются цели и характеристики проекта, определяется его качество и планируются работы.

По нашим наблюдениям, большинство компаний продолжают использовать каскадную модель разработки и внедрения информационных систем предприятия, несмотря на настойчивые рекомендации компаний-вендоров и экспертов в области проектирования и разработки ПО использовать тот или иной вариант итерационной модели! Можно выделить

следующие причины, по которым каскадная модель сохраняет свою популярность:

Привычка. Итерационная модель жизненного цикла появилась относительно недавно, приобрела популярность в последние десять лет, особенно в рамках методологии RUP и MSF. Многие ИТ специалисты получали техническое образование в то время, когда изучалась только каскадная модель, поэтому она используется ими и в наши дни.

Снижение рисков участников проекта (заказчика и исполнителя). Каскадная модель предполагает разработку законченных продуктов на каждом этапе: технического задания, технического проекта, программного продукта и пользовательской документации. Разработанная документация позволяет не только определить требования к продукту следующего этапа, но и определить обязанности сторон, объем работ и сроки, при этом окончательная оценка сроков и стоимости проекта получается на начальных этапах, после завершения обследования. Очевидно, что если требования к информационной системе меняются в ходе реализации проекта, а качество документов оказывается невысоким (требования неполны и/или противоречивы), то в действительности использование каскадной модели создает лишь иллюзию определенности и в действительности увеличивает риски, уменьшая лишь ответственность участников проекта. При формальном подходе менеджер проекта реализует только те требования, которые содержаться в спецификации, опирается на документ, а не на реальные потребности бизнеса.

Есть два основных типа контрактов на разработку ПО, первый тип предполагает выполнение определенного объема работ за определенную сумму в определенные сроки (fixed price), второй тип предполагает повременную оплату работы (time work). Выбор того или иного типа контракта зависит от степени определенности задачи. Каскадная модель с определенными этапами и их результатами лучше приспособлена для типа контракта с оплатой по результатам работы, а именно этот тип контрактов позволяет получить полную оценку стоимости проекта до его завершения. Более вероятно заключение контракта с повременной оплатой на небольшую систему, с относительно небольших весом в структуре затрат предприятия. Разработка и внедрение интегрированной информационной системы требует существенных финансовых затрат, поэтому используются контракты с фиксированной ценой, и, следовательно, каскадная модель разработки и внедрения. Спиральная модель чаще применяется при разработке информационной системы силами собственного отдела ИТ предприятия.

Проблемы внедрения при использовании итерационной модели. В некоторых областях спиральная модель не может применяться, поскольку невозможно использование / тестирование продукта, обладающего неполной функциональностью (например, военные разработки, атомная энергетика и т.д.). Поэтапное итерационное внедрение информационной системы для бизнеса возможно, но сопряжено с организационными сложностями (перенос данных, интеграция систем, изменение бизнес-процессов, учетной политики, обучение пользователей). Трудозатраты при поэтапном итерационном внедрении оказываются значительно выше, а управление проектом требует настоящего искусства. Предвидя указанные сложности, заказчики выбирают каскадную модель, чтобы «внедрять систему один раз».

Встречаются и случаи сознательного самообмана. В одной российской компании, специализирующейся в области разработки заказного программного обеспечения, официально декларируется использование итерационной модели, в то время как на практике во всех проектах применяется каскадная модель, при этом стадии, этапы и рабочие продукты каскадной модели называют терминами итерационной модели из RUP. Можно предположить, что, понимая преимущества итерационной модели, менеджмент хотел бы ее использовать во всех проектах, но по причине сложности, а так же учитывая риски, не может использовать на практике.

Следует добавить, что чем выше уровень доверия между заказчиком и компанией-исполнителем, чем лучше понимает компания-разработчик бизнес заказчика, тем ближе будет выбранная модель жизненного цикла к итерационной модели при прочих равных условиях.

Как уже было сказано, методология проектирования и разработки (совокупность моделей, стандартов, например RUP, MSF, Oracle CDM, etc.) определяет, с помощью каких средств моделирования, в каком объеме будут формализованы требования к информационной системе.
На сегодняшний день существует несколько популярных средств формализованного

описания требований к информационным системам, среди них:

- UML (Unified Modeling Language), Rational Rose или Visual Modeler как инструмент моделирования, сценарии использования для описания работы пользователей (функциональные требования). Use Case диаграммы используются для описания границ системы и выявления функциональных требований самого высокого уровня. Диаграммы деятельности и последовательностей используются для моделирования процессов, а диаграммы классов - для проектирования структуры данных [2].
- SADT (Structured Analysis and Design Technique): язык IDEF (IDEF0, IDEF3), средство моделирования: BPWin / ERWin (после переименования, All Fusion Process Modeler и Data Modeler) [7,8].
- ARIS (Architecture of Integrated Information Systems) доктора Шеера (IDS Scheer AG): нотация eEPC (Event Driven Process Chain) и инструмент ARIS Toolset.

Выбор той или иной совокупности методов зависит от следующих факторов:

- Корпоративный стандарт. В крупных компаниях на определенном этапе возникает необходимость в структурированном описании бизнес-процессов. Построение связанной совокупности моделей обеспечивается за счет создание репозитария объектов (функций, участников процессов, информационных систем, технических средств) обеспечивается за счет использования специализированного средства моделирования, адаптированного к применению определенной нотации. Для обеспечения эффективности инвестиций в приобретение средств моделирования и обучение сотрудников, а так же обеспечения согласованности структурированного описания предприятия (моделирования) компания должна выбрать определенную нотацию и средство моделирования в качестве стандарта, и использовать их как во внутренних проектах, так и в проектах с привлечением консультантов.
- Квалификация персонала. Даже обладая знаниями и навыками работы в различных CASE-средствах, специалист, как правило, отдает предпочтение одному из них, и применяет его наиболее эффективно.
- Информационная система. В зависимости от того, внедряется ли готовая информационная система, либо предполагается разработка заказной системы с использованием объектно-ориентированных средств разработки, используется наиболее подходящая для выбранной системы нотация и средство моделирования. (Например, нотация eEPC и инструмент ARIS Toolset используются в случае внедрения SAP/R3, а UML и Rational Rose при заказной разработке с использованием SQL Server и Visual Studio). То есть CASE-средство должно не только поддерживать нотацию, максимально полно описывающую требования к устанавливаемой системе, но и быть с ней интегрировано.
- Проект. Внедрение информационной системы часто бывает частью проекта по реинжинирингу деятельности компании, одновременно с определением требований к информационной системе проектируется новая организационная структура, формулируются должностные инструкции. В этом случае требования к нотации моделирования и используемым инструментам оказываются шире. В этом случае очевидное преимущество имеет ARIS Toolset, поддерживающий 86 различных нотаций, обеспечивающих моделирование различных аспектов деятельности предприятия. Использование BPWin и нотации IDEF0 так же имеет свои преимущества, поскольку строгость нотации обеспечивает относительно высокое качество модельного описания (связанность моделей, обязательное наличие управления в блоке). Безусловно, только квалификация специалиста позволяет получить результат высокого качества, но правильно выбранная нотация облегчает эту задачу. ARIS является идеальным средством для построения моделей процессов в ходе интервью благодаря простоте и отсутствию жестких требований нотации eEPC.

Функциональные требования к информационной системе, которые описываются, в том числе, и с помощью моделей процессов и структур данных, являются только частью общих требований,

которые содержаться в техническом задании. Раздел требований к информационной системе технического задания может содержать следующие подразделы:

- требования к функциональным характеристикам
- требования к надежности
- настраиваемость
- условия эксплуатации
- требования к информационной и программной совместимости
- требования к документации

Требования к функциональным характеристикам

В этом разделе должны быть указаны требования к составу выполняемых функций, организации входных и выходных данных. При выборе между объектными и структурными методами следует использовать принцип концептуальной общности, который предполагает следование единой философии на всех этапах ЖЦ. Если предполагается использовать структурное программирование, то и на этапе анализа следует использовать структурный подход, а в случае использования объектно-ориентированных языков разработки - объектный анализ и объектное проектирование. При необходимости структурный и объектный подходы могут использоваться одновременно.

Требования к надежности

В разделе должны быть определены требования к обеспечению надежного функционирования: контроль входной и выходной информации, время и механизмы восстановления после программных и аппаратных отказов. В этом разделе описывается организация системы безопасности, включая подсистемы контроля доступа, шифрования и т. п.

Настраиваемость
Определяются требования к адаптационным возможностям ПО, то есть указывается, какие изменения в методах управления и бизнес процессах должны быть предусмотрены.

Условия эксплуатации
В этом разделе описывается необходимое обслуживание, которое требуется для работы системы, например, создание резервных копий, реиндексерование баз и т. п., а так же требования к квалификации персонала (пользователей и обслуживающего персонала).

Требования к составу и параметрам технических средств

Указывается необходимый состав технических средств с указанием их основных технических характеристик. Могут указываться требования к помещениям, в которых будет находиться оборудование. В этом разделе указываются требования к переносимости системы.

Требования к информационной и программной совместимости
Требования к информационным структурам на входе и выходе, методам решения, исходным кодам, языкам программирования и программным средствам, используемым программой.

Требования к программной документации

В этом разделе указывается предварительный состав программной документации, и при необходимости, специальные требования к ней.

Еще раз необходимо подчеркнуть, что состав разделов технического задания определяется особенностями проекта, например, в случае внедрения существующей информационной требования к надежности, информационной и программной совместимости, документации и т.п. имеют номинальное значение, поскольку эти характеристики уже заложены в систему и указываются лишь как часть обязательств в рамках контракта, не влияя на фактический объем работ. В случае разработки заказной системы эти требования необходимо учесть при проектировании, они определять состав работ и структуру проекта.

Динамика изменения требований зависит от выбранной модели жизненного цикла, в каскадной модели требования определяются один раз в начале проекта, а в итерационной - уточняются в ходе выполнения проекта. Во втором случае должна быть предусмотрена процедура управления требованиями. Одним из возможных подходов является представление совокупности требований в виде набора атомарных требований -

утверждений, между которыми выявляются отношения зависимости. Например, продукт Rational RequsitePro позволяет вести базу данных требований, определять их атрибуты, а так же отношения (трассируемость, иерархические связи). При использовании каскадной модели все требования содержаться в техническом задании, затем они преобразуются в архитектурное решение в техническом проекте, в этом случае процедура управления требованиями упрощается, ведь предполагается, что требования не будут меняться в ходе проекта.
Каковы типичные ошибки при определении требований к информационной системе:
неполнота требований (структура). Определяются только часть требований, например функциональные требования, при этом не указываются требования к надежности, производительности, программной совместимости и т.д. применение стандарта на программную документацию (техническое задание) поможет избежать эту проблему. Кроме того, ошибки или неполнота описания бизнес-логики.

Описывается только основной поток процесса, а многочисленные альтернативные потоки не исследуются. При этом количество и сложность альтернативных потоков значительно превосходит количество и сложность основных потоков. Пример: фрагмента основного потока процесса: прибыл заказанный товар на склад, количество и номенклатура совпадают с заказанным, товар отправлен покупателю. Для этого потока существует несколько альтернативных потоков: прибыл заказанный товар, но количество отличается от заказанного (варианты, в большую, меньшую сторону), отличается номенклатура товара (отклонения по размеру, цвету, сортности). Проводится согласование с покупателем. Покупатель согласен (не согласен) получить товар в ином количестве (ассортименте).

Пример можно продолжить, но наша задача лишь показать сложность и количество альтернативных потоков. Выявление альтернативных потоков важно и по той причине, что мониторинг отклонения процесса от основного потока, сбор статистики, является важной функций управления.

- избыточность требований. Избыточность требований встречается так же часто, как и неполнота, как правило, они соседствуют в одном документе. Основные признаки избыточности: описываемые требования реализуются автоматически благодаря используемой технологии разработки или выбранной архитектуре, требования не влияют на архитектуру информационной системы, ее бизнес-логику (например, требования к содержанию данных, вместо требований к структуре и объему информации), требования повторяются многократно в различных частях документа (дублирование). Основная опасность избыточности требований в отвлечении внимания, создании иллюзии полноты выявленных требований.

В каждом отдельном случае разработки и (или) внедрения информационной системы существует оптимальное сочетание модели жизненного цикла разработки, нотаций формализованного описания требований и инструментальных средств. Необходимо учитывать этот факт, в частности, при стандартизации процессов и документации в компании-разработчике программного обеспечения. Если клиенты имеют отличающиеся корпоративные стандарты моделирования бизнес-процессов, различный уровень формализации, используются различные средства разработки, внедряются различные системы, проекты, отличающиеся по масштабу и срокам, необходимо, что бы стандарты предусматривали возможность использования различных моделей ЖЦ, нотаций и инструментов.

Выбор модели ЖЦ, формы представления и уровня детализации требований, CASE-средств, так же значим для успеха проекта по автоматизации, как и выбор информационной системы и компании-поставщика. Если у специалистов компании-заказчика отсутствуют знания, то можно воспользоваться опытом компании-разработчика ИС или привлечь консультантов.

1. Автоматизированные Системы Стадии создания. ГОСТ 34.601-90 Комплекс стандартов на автоматизированные системы. ИПК издательство стандартов, М.: 1997

2. Буч Г. Рамбо. Д. Джекобсон А. Язык UML. Руководство пользователя: Пер. с англ. ДМК, 2000. - 432 с.

3. Гейн К., Сарсон Т. Структурный системный анализ: средства и методы. М: "Эйтекс", 1992 - 274 с.

4. Дж. Мартин. Планирование развития автоматизированных систем. М.: Финансы и статистка, 1984 г.

5. Единая система программной документации. ГОСТ 19.001. М.: ИПК Издательство стандартов , 1998 - 164с.

6. Зиндер Е. З. Соотнесение и использование стандартов организации жизненных циклов систем. Системы Управления Базами Данных · # 3/97 · стр. 41-53

7. Калянов Г. Н. CASE структурный системный анализ. М.: Лори, 1996 - 242 С.

8. Марка Д. А. МакГоуэн К. Методология структурного анализа и проектирования. - М.: МетаТехнология, 1993, 240 с.

9. Ойхман Е. Г. Попов Э. В. Реинжиниринг бизнеса: реинжиниринг организации и информационные технологии. - М.: Финансы и статистика, 1997. - 336 с Скотт Ф. Уилсон и др. Принципы проектирования и разработки программного обеспечения. Учебный курс MCSD. Пер. с англ. - М.: Издательско-торговый дом «Русская редакция», 2000. - 608 с.

10. Уокер Ройс. Управление проектами по созданию программного обеспечения. Издательство «Лори», 2002 г. 424 с.

11. Curran T. Gerhard K. SAP R/3 Business blueprint. - Prentice Hall, 1998. - 287 p.