

МЕТОДОЛОГИЯ ПОСТРОЕНИЯ КОМПОЗИТНЫХ СИСТЕМ ДОКУМЕНТООБОРОТА

1. Введение

На данный момент компьютерные технологии столь глубоко интегрированы в производственные процессы, что чисто бумажные технологии практически перестали существовать. Сегодня во всех процессах документооборота на той или иной стадии используется компьютер. Как результат, влияние информационных технологий, которое изначально было крайне мало, возрастает и уже начинает претендовать на определяющее. Под влиянием первоначально вторичных информационных технологий документооборот непрерывно изменяется, своей новой сущностью заполняет новые ниши, возникающие в человеческой деятельности.

Таким образом, современные решения документооборота надо рассматривать на пересечении электронных и бумажных технологий. Композитная сущность этого представления предопределяет противоречивый характер, две исходные составляющие влияют друг на друга в поиске эффективного решения [1]. Удельное влияние составляющих частей коррелируется уровнем развития технологии и устойчивостью традиций предмета документооборота. Системы документооборота могут успешно функционировать при условии, что будут преодолены противоречия между представлениями о системе заказчика и разработчика, а также будет найден баланс применения бумажных и электронных документов. Помимо этого, наиболее естественным является эволюционный путь создания. Именно данное обстоятельство определяет композитный подход к реализации систем документооборота, где сосуществуют как чисто электронные, так и чисто бумажные документы, а также множество композитных гибридов. Такой подход полностью соответствует принципу смешанного экстремума [1], обеспечивающему, с одной стороны, гармонизацию системы, а, с другой стороны, являющемуся одним из механизмов, определяющих обратную связь эволюционного развития системы.

Понятие «электронный документооборот» (ЭД) является широко используемым и нечетко понимаемым. Большинство специалистов по информационным технологиям считает его определенным и устоявшимся. Тем не менее определения, которые использовались при зарождении понятия ЭД, равно как и современные определения, являются очень общими и детерминированы лишь для высокого уровня абстракций. Например, под определение «документ, информация в якому зафіксована у вигляді електронних даних, включаючи обов'язкові реквізити документа» [2], могут быть подведены любые, даже самые догматично бумажные документы.

Расплывчатость формулировок есть лишь частью целого комплекса проблем индустрии по созданию систем, которые принято называть системами документооборота. Причем интересен тот факт, что принадлежность к этим системам определяется скорее названием, данным автором, чем фактом удовлетворения необходимого списка функциональных потребностей.

2. Постановка проблемы

Жизнь не стоит на месте. Современное общество постоянно изменяется, ставит перед собой и решает новые, все более сложные задачи. Прогресс в технологиях информационного взаимодействия идет темпами, оцениваемыми близко к экспоненциальным [3], что приводит к информационному взрыву.

Процесс становления систем электронного документооборота (СЭД) в современном обществе не только начался и установился, а идет постоянно нарастающим потоком. На базе промышленных, полупромышленных и просто «доморощенных» систем выпускается много систем, которые обеспечивают движение огромного количества документов. Эти документы накапливаются неструктурированной кучей [5], что предопределяет серьезные проблемы на будущее. На данный момент специалистами так и не достигнуто однозначного соглашения, что делать со старыми бумажными документами, каким образом организовать использование старых данных, чтобы не утратить специфических свойств оригинальных документов.

Сегодня, в основном, пользователи и создатели СЭД действуют интуитивно, создавая системы различной степени эффективности и устойчивости, не предусматривающие преемственности. В результате старые постсоветские каноны находятся под сильным влиянием прозападных традиций, что порождает на свет гибриды, отталкивающие потенциальных пользователей этих систем и уступающие их создателям.

Целью данной статьи является создание единой методологии реализации систем документооборота, использование которой позволило бы унифицировать процесс разработки и увеличить эффективность создаваемых СЭД. Приведенная методология является лишь рамочным описанием теоретической и практической работы, основывается на апробированных подходах и может быть использована при проектировании, создании и эксплуатации современных СЭД.

3. Методология

Основу предлагаемой методологии составляют: принципы автоматизации систем организационного управления [4], принципы объектно-ориентированного проектирования [5] и принципы управления проектами [6].

В современном понимании под автоматизацией подразумевается не столько замена людей автоматами, выполняющими их функции, а, в основном, усиление возможностей людей возможностями автоматов. Такая трактовка очень важна, так как одной из основных проблем успешного внедрения новых информационных систем является сопротивление персонала. Люди боятся того, что они станут ненужными или что им придется работать более интенсивно. Поскольку подобные соображения обычно не высказывают открыто, то сопротивление выражается в скрытых формах. Принятие же трактовки усиления [7] позволяет утверждать, что внедрение системы будет означать не сокращение потребности в персонале, а возможность выполнения людьми работы на другом качественном уровне.

Одна из основных идей сформулирована в принципе автоматизации документооборота [4]. Суть этого принципа в том, что СЭД должна проектироваться не только как носитель, а как активный элемент процессов. В объектно-ориентированном анализе это принято называть инкапсуляцией [5]. Не будут успешными реализации, основанные на корректных технических решениях, но не учитывающие существующей практики использования автоматизируемого объекта.

СЭД должна являться не просто посредником, собирающим и обрабатывающим документы, а быть определяющим элементом, без которого использование документооборота не будет возможным. Иными словами, ряд функций должен быть реализован таким образом, что не будет существовать их реализации вне электронной системы.

Этот принцип важен, так как следование ему делает СЭД неотъемлемой составной частью работы организации, что предопределяет привыкание персонала к системе. В любой сложной системе, какой является и СЭД, персонал есть самым слабым, трудно предсказуемым и плохо управляемым звеном. Поэтому важно закладывать такие решения, которые дадут дополнительные рычаги мотивации персонала к работе с системой.

Создание СЭД, как любой процесс созидания, имеет творческое начало. Одна и та же система может быть представлена множеством реализаций, даже если ее проектирует один и тот же человек. В то же время, современные технологии программирования требуют строгой формализации для реализации поставленных задач в заданные сроки и с заданным показателем качества. Апробированные методы достижения поставленных целей требуют четкого предварительного декларирования задач, допускающего последующую, достаточно точную декомпозицию и детализацию.

Для построения современных СЭД необходимо достижение баланса между неконтролируемой спонтанностью творческого начала и формализованной четкостью производственного процесса. При этом нельзя ограничивать креативность, которая позволяет интуитивно находить решения, близкие к оптимальным. В то же время надо обеспечивать устойчивую контролируемость и управляемость процессов разработки. Для решения этой противоречивой задачи принято отдельно выделять ее решения на уровне макро- и микропроцессов.

Макропроцессы определяют общие направления развития организации, выходя за пределы СЭД и сферы ее влияния. Будучи достаточно общими, они слабо подвержены формализации, что придает им неопределенный декларативный характер. Микропроцессы формализуемы, контролируемы и исполняемы.

Это соответствует принципу единства дальних и ближних целей В.М. Глушкова. Данный принцип определяет совместное рассмотрение задач стратегии и тактики с целью получения максимальных успехов в настоящее время с учетом перспектив развития. Дальние и ближние цели отличаются не только временем до их реализации, но и принципом постановки. Ближняя цель должна быть видима и достижима. Ее триггер должен быть достаточно ясным, чтобы констатация достижения была бесспорной, а дальняя цель – достаточно удаленной и указывать скорее направление, чем конкретный путь реализации задач.

С одной стороны, ближние цели являются результатом декомпозиции дальних и являются составными частями будущего результата. В то же время, полная совокупность ближних целей не определяет полностью дальних. Надо сказать, что проблема неполноты составных частей по отношению к общему не является только частной проблемой СЭД, а есть общей методологической проблемой.

Эта ситуация отвечает принципу комплексности задач В.М. Глушкова. Суть его состоит в том, что большинство задач являются комплексными и поэтому не могут быть сведены к простой арифметической сумме мелких задач. Поэтому и решать совокупность декомпозированных задач следует, имея в виду первоначальное целое. Иными словами, между задачами должно происходить постоянное взаимодействие, что превращает их в комплекс, то есть в систему. Помимо этого, на передний план выдвигаются проблемы системного подхода и анализа.

3.1. Макропроцесс

СЭД, как и любая сложная программная система во многом обладает свойством живых организмов эволюционировать. Практика разработки и внедрения программных систем выявила феномен непредусмотренного их использования. Пользователи, как правило, используют систему не совсем так, как первоначально планировалось разработчиками и предполагалось при переделке системы на стадии внедрения.

Казалось бы, проблема этого эффекта состоит в некорректной постановке задачи на стадиях анализа и проектирования. Разработчики, как правило, перекладывают вину на пользователей, считая, что они не понимают основ построения сложных систем. Пользователи, в свою очередь, обвиняют в этом разработчиков, утверждая, что они взялись за постановку задачи без достаточной экспертизы и знаний в данной предметной области.

Таким образом, можно утверждать, что в результате внедрения и эксплуатации система приспособляется к пользователям так же, как и пользователи приспособляются к системе. Невозможно сделать систему, которая была бы сразу полностью адаптирована к требованиям пользователей.

В этой связи можно привести следующий житейский пример: нельзя сделать сразу разношенную обувь. Даже если сделать обувь индивидуально и из самых мягких материалов, тем не менее какое-то, пусть и короткое время, обувь при ходьбе будет вызывать некоторый дискомфорт. Она будет изнашиваться до тех пор, пока нога к ней не привыкнет, а обувь, в свою очередь, примет удобную для ноги форму.

Пример с изготовлением индивидуальной обуви хорош, однако надо учитывать тот факт, что современному индустриальному обществу характерны масштабные промышленные решения. Большинство обуви делается на конвейере по стандартным размерам и стандартным моделям. Этот подход позволяет уменьшить потребительскую цену, но делает обувь менее приспособленной к каждому конкретному потребителю, укладывая наши ноги в прокрустово ложе стандартных размеров.

Опыт современных СЭД показывает, что системы предпочтительно строить, базируясь на промышленных решениях. Причин этому много, но основная состоит в том, что для того, чтобы решения стали более или менее пригодными, им надо пройти долгое тестирование многими миллионами пользователей. Этому требованию могут удовлетворять только те системы, которые всемирно распространены и имеют массовое использование.

Построение СЭД на основе промышленных решений предопределяет использование разработчиком ограниченного набора функциональных возможностей. Полученные путем компоновки решения не удовлетворяют потребностям конкретного пользователя, а являются лишь аппроксимацией к ним. Система модернизируется неоднократно, каждый раз приближаясь к индивидуальным требованиям. Для этого она должна быть достаточно открыта к последующей модернизации. Через несколько итераций адаптации к требованиям пользователя наступает не только привыкание персонала к системе, но и более полное удовлетворение конкретных функциональных потребностей. Это состояние характеризуется тем, что разработчики считают, что удовлетворили все запросы пользователей, а пользователи думают, что, наконец-то, разработчики сделали то, о чем их изначально просили.

Описанная выше адаптация является не только результатом переработки системы, но и результатом изменения объекта автоматизации и декларируемых им задач. В.М. Глушков называет это принципом новых задач. Создание информационных систем является инновационным процессом, внедрением новых технологий. Именно использование новых технологий во многом предопределяет возникновение новых задач, которых не существовало при использовании старых технологий. Системы должны проектироваться с учетом возникновения новых задач, которые возникают, исходя как из сущности новых технологий, так и расширения выполняемых функций.

Макропроцесс, в основном, объединяет процессы в рекурсивном цикле и реализует общий эволюционный цикл доводки системы. Критерии выхода из этой рекурсии индивидуальны для каждого проекта и в основном определяются условиями договора между заказчиком и исполнителем.

3.2. Микропроцесс

Описание микропроцесса опирается на понятие «жизненный цикл» СЭД, которое служит для определения начала разработки и конца внедрения СЭД. Наиболее удобным способом представления процессов реализации СЭД есть представление совокупности процессов в виде проекта, как это определяется в стандарте РМВОК [6].

Жизненный цикл СЭД разделяют на следующие стадии: анализ, проектирование, реализация, внедрение и эволюция. В каждую из стадий входит значительное число процессов, которые объединяются по принципу общности отношений в СЭД. При этом под процессом понимается набор действий, приносящих результат.

Каждый из процессов может быть как одноразовым, так и многократным, за время реализации жизненного цикла СЭД. Это свойство определяется рекурсивной природой создания

сложных систем. При этом основные процессы отличаются от дополнительных тем, что они происходят как минимум хотя бы один раз за время жизненного цикла.

Процессы взаимодействуют между собой как логически, так и информационно. Логическое взаимодействие определяет последовательность происходящих событий, формирует устойчивую сеть взаимосвязей. Информационное взаимодействие определяет информацию, циркулирующую в системе. Можно также говорить, что процессы связаны информацией, которую они производят, так как результат или выход одного является входом для другого.

Процессы – это не детерминированные последовательные события, а чаще параллельные с пересекающимися активностями переменной интенсивности. Начало процессов обуславливается срабатыванием соответствующих триггеров других процессов, находящихся в логической взаимосвязи с ними. Длительность процессов может быть достаточно длительной, что приводит к их прекращению вне зависимости от достижения цели, а по факту завершения проекта.

3.2.1. Анализ

Основная задача анализа – установить основные функциональные требования к создаваемой системе, которые можно выявить на первичном этапе. Впоследствии, на основании выявленных требований, будут строиться первичные сценарии поведения, то есть воспроизводиться предполагаемая реакция системы на внешние раздражители.

Функциональные требования к системе вырабатываются путем совместного рассмотрения требований ключевых участников проекта (stakeholders) [6].

Как показала практика, успех большого проекта в основном определяется мнением об этом факте достаточно узкого круга лиц. Можно определить, что ключевые участники – это лица или организации, активно вовлеченные в проект, чьи интересы могут критично повлиять на успешность проекта. Они также могут быть не активно вовлеченными, но осуществлять надзор и оказывать критичное влияние. Работу по выявлению ключевых участников важно начинать как можно раньше. Многие из участников долгое время остаются в тени, появляясь лишь тогда, когда, по их мнению, дела идут не в том направлении, как они себе представляли.

Требования, формируемые путем опроса ключевых участников, обычно являются противоречивыми. Конфликт интересов есть естественным состоянием любой организации при появлении нового элемента управления, каким является СЭД. Наибольшую опасность представляют те ключевые участники, влияние которых недостаточно для принятия их мнения во внимание, но достаточно для торможения, а то и блокирования проекта. Нахождение компромиссного решения и определение балансов влияний есть слабо формализуемое и очень критичное звено проекта по созданию СЭД.

Разрешение этой проблемы В.М. Глушков сформулировал в [4] как принцип первого лица: система должна заказываться, разрабатываться и внедряться под непосредственным контролем первого руководителя предприятия. Практика проектов показывает, что всякая попытка передоверить решение задачи второстепенным лицам приводит к тому, что созданная система будет решать лишь второстепенные задачи организации. Такой результат является очевидным, так

как система создается на основании анализа требований второстепенных руководителей, которые видят перед собой соответственно второстепенные цели.

В качестве первичного продукта анализа может служить созданный прототип будущей системы. Прототипы служат для раннего обнаружения “тонких мест” и неудачных решений, закладываемых в базис создаваемой системы. Гораздо лучше обнаружить это и закладываемые риски в самом начале, чем потом исправлять их уже в эксплуатирующейся системе.

При этом различают два основных вида прототипов систем: пилотный и модельный типы. Эти прототипы имеют разное предназначение и поэтому отличаются способом и целью их реализации. Пилотный прототип является точным воспроизведением одной из функциональных частей системы. В нем используются промышленные решения, которые будут использованы в рабочей системе. Цель пилотного проекта – показать или проверить какие-то технические решения, которые являются ключевыми и в корректности реализации которых имеются сомнения. Модельный прототип является уменьшенной копией системы или ее части. При реализации модели могут использоваться технологии, отличные от планируемых. Целью модели является получение общего представления о функционировании будущей системы и выработка понимания сути проекта у ключевых участников.

Несмотря на важность прототипа для успешности выполнения проекта в целом, очень важно, чтобы прототип не был эволюционирован в промышленную систему. К сожалению, использование прототипов в качестве промышленного решения очень распространено. Это связано с тем, что руководители, увидев прототип и в целом оценив его полезность, часто принимают решение о внедрении прототипа и прекращении дальнейшего дорогостоящего процесса разработки. Тем не менее, надо понимать, что прототип хоть и является наглядным (что, собственно, и есть его основной задачей), строился совершенно не для эксплуатации. В результате, будучи успешным и экономным решением вначале, он в большинстве случаев потребует дальнейшей дорогостоящей модернизации, что в итоге значительно превысит первоначальную планируемую стоимость системы. При этом становится реальностью поговорка “скупой платит дважды”, и остановка разработки скорее всего приведет как к утрате первоначальных целей, так и к потере инвестированных средств.

Вторая задача анализа – дать описание системы, причем такое, чтобы оно было насколько возможно полным, непротиворечивым, пригодным для восприятия и модификации, измеряемым и управляемым.

На стадии анализа анализируется выбранная система из реального мира, для которой будет построена формальная модель. Анализ, в основном, сосредоточен на изучении поведения реальной системы. Иными словами, главное внимание уделяется тому, что делает система, а не тому, как она это делает.

Поведение системы можно описать с помощью сценариев, которые характеризуются точками ее устойчивых состояний. Точки обозначают хорошо видимые извне и поддающиеся проверке и измерению элементы поведения системы. Такие точки, выявленные на стадии анализа, будем называть функциональными точками системы. В этих точках задачи наиболее просто

формализуются, их синтезирование не представляет значительных трудностей с точки зрения применяемых технологий.

С точки зрения пользователя, функциональная точка представляет некоторое простейшее действие системы в ответ на некоторый простейший “раздражитель”. В простейшем виде можно считать, что функциональная точка представляет отдельную бизнес- функцию конечного пользователя.

С точки зрения аналитика, функциональные точки представляют собой кванты поведения системы. Таким образом, функциональные точки являются мерой сложности системы: чем их больше, тем поведение системы более сложное.

Семантика поведения системы, то есть семантика функциональных точек, задается сценариями. Каждый сценарий представляет одну функциональную точку. Первичные сценарии используются для иллюстрации ключевого поведения, вторичные – для описания поведения в исключительных ситуациях.

Если воспринимать моделируемую систему и ее внешние раздражители как совокупность сложноорганизованных объектов, описание взаимодействия которых дает в качестве результата формальную модель, то такое восприятие отвечает системному анализу объекта и системы управления им, названное В.М. Глушковым принципом комплексного (системного) подхода [4].

На стадии анализа целесообразно выделяют участки, наиболее пригодные для автоматизации. Начинать следует с наиболее простых участков, так как удачное начало проекта во многом предопределяет дальнейшее удачное продолжение и завершение. Такие участки характеризуются, прежде всего, высоким уровнем готовности персонала, подходящим техническим оснащением и заинтересованностью руководства в работе данного участка.

Считается полезным также привлекать службу качества организации (предприятия) для работы по разработке сценариев. По сути именно сравнение сценариев с реальным поведением системы определяет качество полученной модели, то есть ее адекватность системе из реального мира.

Полученные в результате анализа данные объединяются в один формальный документ, который формулирует требования к системе, полученные в результате анализа - Техническое задание на СЭД.

3.2.2. Проектирование

Основная задача проектирования – создание архитектуры будущей системы и разработка плана, достаточного для последующей ее реализации и внедрения. Процесс проектирования целесообразно начинать сразу после построения некоторой приемлемой формальной модели. На стадии проектирования описанные функциональные точки обследуются не только аналитиком, но и архитектором, и технологом. Совместная работа этих специалистов дает архитектуру системы, описанную в проектных решениях.

При этом не следует начинать проектирование центральных узлов до окончания этапа анализа. И в то же время нельзя затягивать начало проектирования в ожидании завершения этапа

анализа. Построение модели, будучи процессом итеративным, может последовательно уточнять модель поведения системы, пытаясь получить в определенном смысле идеальную, а, следовательно, в большинстве случаев недостижимую модель. Такая ситуация классифицируется практиками как паралич анализа.

При проектировании полезно закладывать возможность реализации максимального количества функциональных потребностей, выявленных при анализе. При этом целесообразно иметь несколько разных решений для одной и той же задачи. Такой подход назван В.М. Глушковым принципом функциональной избыточности [4].

Пользователи часто используют систему не совсем так, как это предусматривали разработчики. Поэтому важно, чтобы проектируемая архитектура содержала достаточные возможности для адаптации. Жестко определенные функциональные возможности потенциально предопределяют последующие конфликты во время внедрения и эволюции системы. Желательно, чтобы система имела функциональный модуль либо модули, которые позволяют производить изменение конфигураций функционирования. Таким образом, пользователь может получить некий каркас, “обшитый” устойчивыми решениями и в последующем произвести адаптацию исходя из реалий эксплуатации. Это соответствует тому, что В.М. Глушков формулирует как принцип гибкости системы [4].

Другой стороной описанной адаптивной специализации является типизация. Современное программное обеспечение изготавливается в крайне сжатые сроки с учетом ограниченных требований постановщиков. Ушли в прошлое времена, когда система годами отлаживалась до запуска в промышленную эксплуатацию. В то же время безответственным и авантюристичным является построение серьезных и сложных систем, если опираться лишь на надежду и необоснованную уверенность в том, что они будут работать.

В рамках АСУ В.М. Глушков описал решение данной проблемы как принцип разумной типизации. Выходом является построение индивидуальных систем из модулей хорошо отлаженных промышленных систем. Это соответствует использованию крупноблочной сборки, где крупные блоки сами являются системами либо подсистемами, используемыми в промышленной эксплуатации миллионами пользователей. Применение в качестве каркаса “доморощенной” системы приводит к получению закрытого сложно адаптируемого решения, что, в итоге, делает решение краткосрочным и в свою очередь приводит к потере инвестированных в него ресурсов.

Рациональное отношение к потребляемым ресурсам требует также использования совместимых решений. Под совместимыми решениями понимается возможность взаимодействия различных уровней системы для движения информации.

На эту проблему обратил внимание В.М. Глушков в [4] и определил это как принцип единства информационной базы. Информация, которая образуется на разных уровнях системы, используется по-разному для решения разных задач. Для этого необходимо использовать интерфейсы в том смысле, как это понимается при объектно-ориентированном проектировании. Использование множественных интерфейсов для обработки единого информационного массива позволяет удовлетворять множественные запросы пользователя и в то же время сохранять целостность данных. Таким образом, в нашей технологии принцип В.М. Глушкова дополнен

полиморфизмом представления данных, что позволяет решать современные задачи, характеризующиеся большими объемами и слабой детерминированностью.

Одна из основных задач – создание “правильной” архитектуры будущей системы, которая обеспечит ее концептуальную целостность. Стратегическое направление обеспечивается целенаправленным тактическим движением. Архитектура должна обеспечивать баланс между стратегическим и тактическим решениями.

Идеальная архитектура практически недостижима, но можно достаточно четко описать свойства, присущие сильной архитектуре:

- архитектура представляет собой многоуровневую систему абстракций. На каждом уровне абстракции взаимодействуют между собой по формализованным протоколам. Абстракции имеют четкие интерфейсы для внешнего мира и основываются на хорошо продуманной реализации;

- система абстракций имеет слабое зацепление, функциональную связность, достаточность, полноту и предельную примитивность;

- на каждом уровне интерфейс абстракций строго ограничен от реализации. Реализация может быть изменена. При этом интерфейс должен остаться неизменным. Таким образом, изменяясь внутренне, абстракции продолжают соответствовать внешним ожиданиям, то есть своему протоколу;

- архитектура проста в понимании, прозрачна и приспособлена к масштабированию.

Вторая задача проектирования – разработка и согласование детального плана реализации и внедрения. В современных условиях задачи всегда реализуются в ограниченные сроки и при ограниченных ресурсах. Это связано с тем, что пользователю не нужно решение, которое займет больше времени, чем у него отведено и не нужно решение, которое потребует ресурсов больше, чем он может выделить.

Разработка плана – это процесс создания целостного и непротиворечивого документа, который может быть использован для исполнения и контроля. Фактически в плане описываются с большей или меньшей степенью детализации действия и мероприятия, которые будут реализованы на стадии реализации и внедрения.

Процесс разработки плана имеет итеративную природу и движется от первоначального черновика, содержащего грубые оценки по срокам и ресурсам, к конечному списку работ с определенными стоимостями и сроками. В процессе разработки плана получается упорядоченный список работ, который называют Иерархической Структурой Работ – ИСР (WBS – Work Breakdown Structure) [5].

Каждый элемент ИСР ставится в соответствие свойствам и ресурсам, необходимым для выполнения данного элемента. В частности, элементу соответствуют сроки, стоимость и квалификация персонала.

Поскольку ИСР является результатом декомпозиции конечного результата с точки зрения достижения цели, то список работ является конечным и полным. По сути, выполнение всех работ, предусмотренных в плане, означает достижение цели проекта, то есть создание СЭД.

3.2.3. Реализация

На стадии реализации на основании проекта изготавливается конкретная программная реализация. Таким образом, из общей формальной модели получается конкретная формальная модель. Конкретная модель является функционально подобной системе реального мира, то есть реализовывает сценарии ее поведения.

По сути, реализация является лишь исполнением плана, разработанного на стадии проектирования. Именно на стадию реализации приходится большая часть времени от общего времени проекта. Большая часть ресурсов затрачивается, как правило, тоже на стадии реализации.

Исполнение плана – это координация персонала и управление потреблением ресурсов с целью достижения поставленной цели. Персонал проекта включает в себя персонал заказчика и сотрудников заказчика, которые при этом не только консультируют, но и обучаются.

Исполнение необходимо постоянно сравнивать с планом, выявляя отклонения. Как правило, все планы выполняются с отклонениями. Создание идеального плана невозможно. Следовательно, возникновение отклонений неизбежно. Если даже удалось создать план, который обеспечит нужный результат, то все равно за время разработки, как правило, происходит изменение первоначальных требований клиента, что приводит к изменению плана.

Отклонения анализируются с точки зрения их позитивного или негативного влияния на вероятность достижения цели. Изменения, реализация которых будет иметь позитивное влияние, называются возможностями, негативное – угрозами. Управление рисками в реализации сводится к мониторингу процесса с целью уменьшения угроз и увеличения возможности. Основными характеристиками рисков являются вероятность и степень их влияния в случае возникновения.

Реализация – итеративный процесс, на каждой итерации которой возникает некоторое приближение – модель, поведение которой все больше и больше соответствует описанным сценариям. Степень реализованности сценариев определяет адекватность модели. Промежуточные модели принято называть релизами или версиями. Важно обеспечивать преемственный характер изменений, так как более успешным всегда являлся эволюционный подход, а не революционный. Необходимо помнить, что процесс устранения ошибок есть процесс внесения новых ошибок, и каждая модернизация системы, кроме положительных последствий, обязательно будет иметь и негативные.

Имеет смысл некоторые участки реализации запускать в пробную эксплуатацию еще на стадии разработки. Это позволяет выявить несовместимые решения, принятые на этапе проектирования и изменить проект и его реализацию до стадии внедрения, тем самым избежав значительных финансовых, временных и человеческих затрат.

3.2.4. Внедрение

На стадии внедрения происходит установка и запуск изготовленной программной модели в промышленную эксплуатацию. Эта стадия, как отмечалось выше, характеризуется сильным сопротивлением персонала. Люди, в своем большинстве, опасаются внедрения систем автоматизации, подозревая грядущее увеличение объема работ и возможное их увольнение.

На этой стадии важно продемонстрировать персоналу, что система является их помощником, а не заменой. Для этого полезно совмещать электронные и неэлектронные части рабочих процессов на одних и тех же исполнителях. То есть документ в своем физическом и виртуальном движении обрабатывается одним и тем же участником. В.М. Глушков называет это принципом совмещения подготовки документов [4]. В современных системах документооборота редко встречается бумажная форма, не имеющая электронного соответствия, и наоборот. Поэтому имеет смысл во всех точках, где создаются бумажные документы, предусмотреть возможность инициации в системе их электронного аналога.

При внедрении всегда встает вопрос о начальном наполнении системы. Редко встречаются случаи, когда СЭД создается в самом начале деятельности организации и ее использование не требует накопления больших объемов бумажных документов. Чаще всего при внедрении рассматривается вопрос поэтапного ввода в систему документов, преобразуемых к новым электронным форматам. Причем часто вырабатывается решение, в котором в СЭД используется только электронный каталог, содержащий ссылки на место хранения бумажных документов.

Сложность этого вопроса определяется тем фактом, что самые большие проблемы информационных систем возникают в звене взаимодействия человека и компьютера. Человеческое и машинное представления сильно отличаются друг от друга, а процесс преобразования от одного вида к другому – технически сложный и неоднозначный. В.М. Глушков называет его принципом минимизации ввода-вывода. Это объясняется тем, что именно на процессах ввода-вывода возникает наибольшее число ошибок и происходят самые невероятные сбои.

Внедрение должно быть реализовано таким образом, чтобы возможности вернуться к старой системе уже не существовало.

3.2.5. Эволюция

Системы электронного документооборота и предмет их автоматизации, как уже отмечалось выше, обладают некоторыми свойствами живых организмов, в частности, свойством эволюционировать. За время разработки СЭД его предмет значительно изменяется. При этом часто меняются декларируемые цели и поставленные задачи. Во время внедрения система становится непохожей на ту модель, которую разрабатывали при проектировании.

Эволюция – одна из самых длительных и определяющих стадий создания СЭД. Можно определить стадию эволюции как период с момента начала внедрения и до окончания использования системы хотя бы одним участником. За это время система до неузнаваемости меняет свои интерфейсы и реализацию и, зачастую, о преемственности можно догадаться только по названию.

Именно на стадии эволюции появляется продукт, который отвечает реальным интересам пользователя. Это связано с тем, что в промышленной эксплуатации пользователь определяет,

какие функции нужно развить, а какими – пренебречь. Поэтому важно, чтобы система была открытой, простой в понимании модернизации.

Это формулируется принципом постоянного развития, который состоит в том, что и программная и реальная системы находятся в постоянном развитии. Поэтому необходимо обеспечить синхронность этих процессов. СЭД должна развиваться вместе с организацией, чтобы инкапсулированная на этапе внедрения система стала естественной и неотъемлемой частью организации.

4. Выводы

В статье предложена интегрированная методология реализации композитных систем документооборота, которая объединяет в себе три базовых подхода: принципы проектирования АСУ В.М. Глушкова, объектно-ориентированный подход при реализации программных продуктов и принципы управления проектами PMBOK.

В соответствии с принципом смешанного экстремума [1] целесообразно композитное сочетание электронного и бумажного документооборота, что делает практическое использование системы более эффективным и уменьшает сопротивление персонала при ее внедрении.

Изложенный подход к улучшению функционирования систем документооборота соответствует так называемому стратегическому подходу Р. Беллмана [8]. Когда вместо того, чтобы строить оптимальную последовательность решений от фиксированного состояния системы, определяется оптимальное решение любого состояния системы. Такой подход приводит к аппроксимациям в пространстве стратегий в том смысле, что каждое последующее приближение улучшения функционирования системы, улучшает результат предыдущего.

Принятый достаточно высокий уровень абстракции делает методологию справедливой для широкого круга задач. В то же время сформулированные в статье принципы служат каркасом, который позволяет при наращивании сохранить целостность предложенной модели.

Разработка теоретических основ композитного документооборота является крайне актуальной для современного управления тематикой и открывает перспективы для дальнейшего теоретического развития и практического внедрения полученных результатов.

Данная методология может быть использована в качестве базиса для разработки нормативной базы СЭД отраслей и предприятий.

СПИСОК ЛИТЕРАТУРЫ

1. Теслер Г.С. Принцип смешанного экстремума как основа эволюции вычислительных средств // Математичні машини і системи. – 2002. – № 1. – С. 3–13.
2. Закон України про електронні документи та електронний документообіг // Відомості Верховної Ради (ВВР). – 2003. – № 36. – С. 275.
3. Engelbart D.C. A Conceptual Framework for the Augmentation of Man's Intellect. – London: VI Spartan Books, 1963.
4. Глушков В.М. Введение в АСУ. – К.: Техніка, 1972. – 312 с.
5. Booch Grady. Object-Oriented Analysis and Design with Applications. – Third Edition, Addison-Wesley, 1993.
6. Project Management Institute. A guide to the Project Management Body of Knowledge. – 2000.

7. Engelbart D.C. Toward Augmenting the Human Intellect and Boosting our Collective IQ // Communications of the ACM, 38: 8. – 1995. – August. – P. 30 – 33.
8. Беллман Р. Теория регулирования // Математика в современном мире. – М., 1967. – 173 с.