

РАБОТА С ТРЕБОВАНИЯМИ И ОСОБЕННОСТИ РАЗРАБОТКИ ПРИ ВНЕДРЕНИИ СИСТЕМ АВТОМАТИЗАЦИИ УЧЕТА

О.В. Максименкова,

преподаватель Государственного университета — Высшей школы экономики,
e-mail: omaksimenkova@hse.ru.

Ю.Ю. Овчинников,

ведущий специалист по внедрению, Внедренческий центр «1С-Рарус»,
e-mail: ovsy@rarus.ru.

Адрес: 129327, г. Москва, ул. Чичерина, д. 2/9 – 337.

Ввиду того, что все предприятия ведут учет схожим образом, появилась большое количество типовых тиражных решений по автоматизации бизнес-процессов учета. Вследствие этого, очертился спектр задач по интеграции некоего дополнительного функционала в тиражное решение. Одним из насущных вопросов разработки программного обеспечения становится внедрение и разработка в рамках систем автоматизации учета.

Ключевые слова: система автоматизации учета, автоматизация учета, учет, работа с требованиями.

Введение

Под бизнес-процессом будем понимать связанную совокупность функций, в ходе выполнения которой потребляются определенные ресурсы, и создается продукт, представляющий ценность для потребителя [1]. Автоматизация бизнес-процессов — это инвестиционная деятельность, которая может быть направлена на увеличение эффективности работы предприятия, инвестиционной привлекательности, эффективности управления предприятием и т. д. Ясное понимание бизнес-процессов предприятия крайне важно для создания программного обеспечения, которое будет полезно заинтересованным лицам

[2]. Заинтересованными лицами являются заказчики проекта автоматизации, например, собственники предприятия.

Особенности выделение клиентов и сбор требований при автоматизации учета

Заказ на автоматизацию поступает, как правило, от руководства фирмы. Заказчиком здесь и далее будем считать предприятие, а клиентом — носителя бизнес-функции, деятельность которого автоматизируется. Например, кассир является клиентом. Бизнес-функцией клиента является совокупность бизнес-операций, сгруппированных по определенному признаку, бизнес-операцией является эле-

ментарное (неделимое) действие, выполняемое на одном рабочем месте [1]. Таким образом, специалист по автоматизации имеет дело с большим количеством клиентов, каждый из которых является представителем своей узко-профильной предметной области.

Клиент, в свою очередь, ожидает от специалиста или аналитика достаточно глубокого понимания своей предметной области. Однако в роли аналитика в процессе автоматизации учета нередко оказываются специалисты по внедрению (разработчики), не владеющие системными, детальными знаниями по всем профилирующим областям знаний, связанных с учетом.

Обеспечить успешное проектирование в такой ситуации могут методы проектирования на основе предметной области (Domain-Driven Design (DDD)). Разработчику следует узнать как можно больше о предметной области, с которой ему приходится работать, и привлечь к проекту пользователей, чтобы они могли не только передать свой опыт работы в предметной области, но активно и конструктивно участвовать в самом проекте [3].

Клиент, выступающий в роли эксперта предметной области, не обязан быть программистом. Его задача — хорошо знать все аспекты исследуемой проблемы и свободно оперировать понятиями из словаря предметной области [4].

Таким образом, работа по автоматизации учета ведется в тесном сотрудничестве с клиентами.

Одним из наиболее важных моментов проектирования на основе предметной области является первичный сбор требований.

Как уже указывалось выше, процесс взаимодействия с клиентами должен выстраиваться специалистом по внедрению учетной системы. То есть успех взаимодействия определяется, в том числе, и методами сбора и анализа информации.

Рассмотрим наиболее простой случай устранения неполадок в системе. Здесь, например, эффективным является метод сбора информации известный как «наблюдение за действиями пользователя» (shadowing)[5]. Специалист по внедрению учетной системы отслеживает действия клиента, приводящие к ошибке или некорректной ситуации в системе.

В случаях, когда информация, полученная от клиента, необходима для доработки старого или проектирования нового функционала, эффек-

тивным оказывается метод интервьюирования (interviewing).

Обобщенную схему такого взаимодействия между клиентом и специалистом по автоматизации и учету можно описать следующим образом: разработчик получает от клиента первоначальный сбор требований. Стоит отметить, что клиент, при этом, сам может не отдавать себе отчета в том, что его требования не соответствуют его желаниям. Тем не менее, разработчик выполняет разработку чернового варианта приложения (прототипа) и демонстрирует клиенту. Если ожидания клиента удовлетворяются, то происходит «доводка» системы. В противном случае, прототип дорабатывается сообразно уточненным требованиям клиента.

В некоторых случаях итеративным становится еще этап сбора требований. В процессе интервьюирования клиент может осознать, что не имеет ответов на все заданные ему вопросы. В таком случае встречи повторяются. На каждом этапе итеративного сбора требований появляются новые ответы и уточняются старые; задаются новые или уточняются старые вопросы. Путем последовательных итераций формируется представление о том, что клиент хочет получить в процессе автоматизации учета, утверждаются или изменяются цели или, как вариант, происходит полный отказ от реализации задачи, например, в случае ее нереализуемости или внутренней противоречивости. К несомненному достоинству метода итеративного сбора требований следует отнести глубокую вовлеченность клиента в процесс проектирования и, как следствие, понимание им того или иного исхода проектирования.

Задачу анализа требований облегчает получение от клиента некоторых артефактов. С точки зрения сбора информации артефакт — это любой объект, физически существующий в бизнес-среде, описывающий элемент основного бизнес-процесса [5]. При разработке в рамках внедрения учетных систем артефактами являются полученные от клиента отчетные формы. Отчет практически в готовом виде задает структуру хранения учетных данных.

Также важным артефактом при внедрении и разработке учетных систем являются должностные инструкции (если они существуют на предприятии). Должностные инструкции могут быть практически без изменений преобразованы в набор ролей пользователей в системе [6].

Некоторые особенности разработки и внедрения систем автоматизации учета

Специфика автоматизации ведения учета заключается в том, что все предприятия ведут учет схожим образом. Поэтому существует большое количество типовых тиражных решений. Однако на каждом предприятии ведение учета имеет свои особенности. Например, предприятие может предоставлять сотрудникам отпуск/отгул/отпуск за свой счет на полдня. С точки зрения законодательства в учете отражается только целодневное отклонение от рабочего графика. Другой пример: предприятие использует специфический алгоритм распределения транспортно-заготовительных расходов на себестоимость материально-производственных запасов в зависимости от объема. В таком тиражном решении, как 1С: Управление производственным предприятием, например, реализованы только алгоритмы по весу, количеству или сумме. Следовательно, возникает необходимость доработки типового решения. То есть работа по автоматизации учета отличается, в данном случае, от разработки программного обеспечения тем, что полученный от пользователя срез требований необходимо соотнести с функционалом выбранного в качестве базового тиражного решения.

Кроме того, многие предприятия совершают специфические операции неавтоматизированные в типовых решениях. Например, лизинговые операции. В таком случае работа по автоматизации учета может состоять не только в реализации дополнительных алгоритмов на базе существующих решений, но и в создании полноценных дополнительных программных компонент с учетом требований отдельных конечных клиентов и заказчика в целом.

Неотъемлемой частью процесса внедрения учетной системы на предприятии является этап ввода начальных остатков (остатков товаров на складах, денежных средств на расчетных счетах и в кассе, остатков дебиторской и кредиторской задолженностей и т.д.). Так же проблема ввода начальных остатков, как правило, встает при разработке нового функционала или модификации существующего в уже работающей системе.

Одной из особенностей учетных систем является то, что большинство учетных данных являются периодическими, то есть не имеют смысла без определения даты, на которую они актуальны. Например, говоря об объеме продаж, мы обяза-

тельно указываем период, за который этот объем был достигнут (оборотный показатель). Говоря об остатке товаров на складе, мы обязательно укажем на какую дату этот остаток товаров был актуален (остаточный показатель). При регистрации факта поступления средств на расчетный счет, обязательно зафиксирована дата, когда это произошло. Следует отметить, что в практике учета распространена регистрация фактов хозяйственной жизни так называемым «задним числом», когда событие, произошедшее в прошлом, регистрируется в настоящем времени.

Также важной задачей автоматизации учета является обеспечение обратной совместимости системы. Обратная совместимость – это такой способ реализации изменений в системе, при котором для работы с данными, относящимися к прошлым периодам, используются алгоритмы, актуальные на тот период.

Предположим, что у нас есть учетная система, содержащая объект для регистрации факта поступления товара на склад фирмы, объект для регистрации факта продажи товара и объект для хранения значений остатков товаров в разрезе складов. В какой-то момент у руководства фирмы возникло желание видеть товарные остатки в разрезе подразделений фирмы, закупивших этот товар (К примеру, появилась необходимость распределять складские расходы между подразделениями). В объект для хранения остатков добавляется реквизит «Подразделение», в операцию поступления товаров так же добавляется подразделение и прописывается заполнение соответствующего реквизита в товарных остатках. В объекте, регистрирующем продажу у нас, предположим, уже существовал реквизит «Подразделение» – он использовался для формирования отчета по продажам в разрезе подразделений. В этом объекте прописывается алгоритм списания остатков в разрезе подразделений. Далее необходимо на некую дату разнести существующие остатки по подразделениям и начать пользоваться модифицированной системой. Но если после ввода системы попытаться перепровести задним числом продажу, система, в соответствии с актуальным алгоритмом спишет остаток товара с того подразделения, от которого оформлена продажа. В результате этой операции на таком подразделении образуется отрицательный остаток товаров. Для исключения подобной ситуации, необходимо реализовать некий дополни-

тельный контроль: если документ относится к прошлому периоду, то и проводиться он должен по старым правилам.

Приведем другой пример.

Предположим, в системе реализована функция печати документов по некоей регламентированной законодательством форме, и приходит день, когда законодатель принимает решение изменить данную форму. В связи с этим, необходимо реализовать в системе новую печатную форму документов и обязательно предусмотреть возможность печати старых документов по старой форме (если, к примеру, понадобится перепечатать документы за прошлые периоды). Из приведенных примеров видно, что для каждой версии системы есть период актуальности и, модифицируя функционал, необходимо обеспечивать его корректную работу для данных, относящихся к старым периодам.

Заключение

Таким образом, в процессе сбора требований при автоматизации бизнес-процессов учета оказываются эффективными не только классические методы проектирования, но и их модификации, такие как итеративный сбор требований до начала проектирования.

Процесс разработки, в свою очередь, имеет следующие особенности:

- ✦ разрабатываемое решение необходимо встраивать в некое тиражное;
- ✦ необходимо реализовывать дополнительный функционал, обеспечивающий ввод начальных остатков;
- ✦ необходимо обеспечивать корректную работу модифицированного функционала со старыми данными.

Внимание к выделенным особенностям позволяет избежать большого количества типовых ошибок при внедрении и доработке функционала учетных систем. ■

Литература

1. Калянов Г. Н. Моделирование, анализ, реорганизация и автоматизация бизнес-процессов. — М.: Финансы и статистика, 2006.
2. Р. Деннис Гиббс. Управление проектами с помощью Rational Unified Process. Практические советы. — М.: КУДИЦ-ПРЕСС, 2007.
3. Д. Нильссон. Применение DDD и шаблонов проектирования: проблемно-ориентированное проектирование приложений с примерами на С# и .NET. — М.: ООО «И.Д. Вильямс», 2008.
4. Г. Буч, Р. Максимчук, М. У. Энгл, Б. Д. Янг. Объектно-ориентированный анализ и проектирование с примерами приложений. — М.: ООО «И. Д. Вильямс», 2008.
5. Microsoft Corporation. Анализ требований и создание архитектуры решений на основе Microsoft .NET. Учебный курс MCSD. — М.: Издательско-торговый дом «Русская Редакция», 2004.
6. Хомичевская В. Н. 1С: Предприятие 8.0. Управление производственным предприятием. Ведение бухгалтерского учета. Самоучитель. — СПб.: БХВ-Петербург, 2006.



*Издательство «Техносфера»
пополнило серию «Мир программирования»
новой книгой*

Виктора Александровича Сердюка,
*преподавателя кафедры управления
разработкой программного обеспечения
ГУ-ВШЭ*

*и генерального директора ЗАО «ДиалогНаука»
«Новое в защите от взлома
корпоративных систем».*