**Методика создания программного обеспечения для систем управления предприятиями с использованием типовых программных компонентов**

С.Трофимов

Разработка и внедрение сложных программных комплексов для предприятий хранения и переработки зерна требует привлечения значительных ресурсов как разработчика так и самого предприятия, поэтому стоимость таких решений для промышленного предприятия среднего размера начинается от нескольких десятков тысяч долларов в нижнем ценовом диапазоне и практически не имеет верхней ценовой границы. Возникает вопрос есть ли пути ее уменьшения со стороны разработчиков системы.

Стоимость совокупного владения программной системой складывается из стоимости создания, стоимости внедрения и стоимости сопровождения. Причем достаточно распространенной ошибкой при выборе ПО является учет стоимости только самой системы. Тогда как затраты на внедрение и сопровождение автоматизированной системы управления предприятием (АСУП), которая должна работать десятилетиями, значительно перекрывают стоимость самого ПО.

В настоящее время методы и средства уменьшения трудоемкости разработки и сопровождения ПО (а значит и его стоимости) разработаны достаточно глубоко. Это такие методы программной инженерии, направленные на создание сложных систем, как разработка открытых систем [1], разработка принципов оптимального разбиения систем на модули [2], методы объектно-ориентированного анализа и проектирования систем [3]. Сейчас для разработки ПО различного назначения широко применяется повторное использование программных объектов и программных компонентов [4], предлагаемых производителями средств ускоренной разработки приложений (rapid application development RAD). Использование CASE-средств [5] также позволяет значительно сократить трудоемкость проектирования и сопровождения систем.

Казалось бы, к вышесказанному уже нечего добавить, и при использовании перечисленных методов и средств стоимость создания и сопровождения ПО будет минимальной. Однако это не так. Резервы кроются в определенной структуре ПО системы, при помощи которой осуществляется снижение стоимости внедрения и сопровождения ПО АСУП. Сразу хочу заметить, что “серебряной пули” по выражению Ф. Брукса [6], позволяющей разрабатывать легко сопровождаемые, дешевые программные системы, еще не создано, и предлагаемый подход требует дальнейших исследований, однако, он уже применялся в той или иной степени при разработке Автоматизированной информационной системы для комбинатов хлебопродуктов (АИС КХП) которая успешно работает в режиме промышленной эксплуатации на нескольких десятках предприятий отрасли.

Особо хочу отметить, что речь идет именно о структуре программного обеспечения, а не собственно информационной системы. Не для кого не секрет, что даже жестко заданную функциональность можно реализовать различными методами, причем не всегда рассчитанными на дальнейшее сопровождение. Здесь можно отметить особенности реализации при помощи различных языков программирования и личные предпочтения того или иного программиста. Последнее часто доминирует при слабом контроле или отсутствием такового со стороны руководителя проекта. Многие решения в этом случае отдаются на откуп программисту, который часто не имеет должной квалификации для создания действительно легко сопровождаемой и дешевой программной системы.

Это при том, что структуру программного продукта трудно определить по внешнему виду. Система отвечает заявленной функциональности и неизвестно, используется ли в создаваемой системе та или иная конструкция языка, создается ли система на основе взаимодействующих объектов или написана при помощи условных и безусловных переходов.

Трудно, да часто и невозможно контролировать весь программный код системы. И уже через какое-то время работы выяснятся, что для того чтобы внести изменения в код, написанный программистом-контрактником, заинтересованным в быстром создании необходимой функциональности, необходимо частично или полностью переписать недокументированный, неструктурированный, написанный без учета дальнейшего сопровождения код.

Мне можно возразить, что для этого и существуют CASE-средства, однако, это только средства, а структура ПО создается все-таки программистом, в лучшем случае под руководством проектировщика.

И даже в этом случае процесс разработки осуществляется без использования неких общих законов, как, например, законов физики в строительстве, и полностью основывается на методах проб и ошибок [7]. Таких законов для разработки ПО просто не существует, потому что научные основы для создания ПО только разрабатываются.

Таким образом, предоставление программисту готовой структуры ПО, построенной на основе объективных исследований конкретной предметной области, и методических рекомендаций по созданию ПО на основе такой структуры может значительно уменьшить трудоемкость создания и, что немаловажно, сопровождения системы и уберечь от дорогостоящих логических ошибок при создании программных модулей.

Термин “типовой программный компонент” определяется как набор программных объектов, учитывающих особенности предметной области, неотделимых друг от друга в рамках выполнения определенного класса задач, имеющих унифицированный интерфейс, позволяющий ТПК взаимодействовать между собой без дополнительного координирующего кода и имеющих возможность изменения своих свойств без внесения изменений в программный код.

Основой ТПК при таком подходе является динамически создаваемое представление данных, использующее для этого хранимую в профиле пользователя информацию. Эти мета-данные представляют собой описатели обрабатываемых полей таблиц, шаблонов ввода-вывода и ссылок на другие таблицы. При этом все алгоритмы вывода на печать, поиска, группировки, добавления, удаления записей реализованы независимо от обрабатываемой информации. Для заполнения ссылочной части таблиц предусмотрен интерфейс для взаимодействия с тем же модулем, но настроенным для работы с другими данными, т.е. предусматривается рекурсивный вызов.

Для создания ПО на основе ТПК предлагаются следующие шаги:

1.Создание ТПК - наиболее трудоемкий и дорогостоящий этап, который должен быть выполнен квалифицированным программистом. Он включает в себя следующие пункты:

выбор средства создания (RAD,CASE);

определение необходимого набора функций;

определение необходимого объема адаптации;

создание иерархии классов;

создание исполняемого кода ТПК.

2.Анализ предметной области для дальнейшего создания модулей системы на основе ТПК. Этот этап выполняется аналитиком системы и не включает в себя непосредственного программирования. Его можно разбить на следующие пункты:

определение набора информационных журналов для работы ТПК;

определение структуры информационных журналов, форматов полей данных и связей;

определение функций и квалификации пользователей.

3.Модулей системы на основе ТПК и профилей пользователей. Этот этап выполняется программистами средней квалификации и включает основной объем работ по созданию системы. Этот этап разбивается на следующие пункты:

создание структуры для динамического формирования представлений данных и при необходимости программных модулей для их обработки на основе шаблонов ТПК;

заполнение структуры для динамического формирования представлений данных согласно структурам информационных журналов, форматам полей и связей;

предварительное заполнение профилей пользователей согласно функциям и квалификации пользователей.

4.Настройка ПО осуществляется отделом внедрения и, в дальнейшем, отделом сопровождения непосредственно на предприятии и включает в себя следующие пункты:

определение конфигурации дисковых устройств для конкретных рабочих станций;

заполнение эталонов адресов хранения файлов;

окончательная настройка профилей пользователей согласно функциям и квалификации конкретных пользователей с настройкой форм ввода-вывода.

Предложенная методика позволяет сократить расходы на создание и сопровождение ПО АСУП путем сокращения размеров программного кода требующего дальнейшего сопровождения, повышения его гибкости, что позволяет осуществлять основную адаптацию к изменяющимся условиям эксплуатации силами самого предприятия без привлечения разработчика. Снижаются требования к квалификации программистов, повышается надежность системы за счет использования многократно протестированных компонентов.

**Список литературы**

 [1] Липаев В.В., Филинов Е.Н. Мобильность программ и данных в открытых информационных системах. –М.: Научная книга, 1997. – 368 с.

[2] Мамиконов А.Г. Методы разработки автоматизированных систем управления. –М.: Энергия, 1973. – 336 с.

[3] Буч Г. Объектно-ориентированный анализ и проектирование с примерами приложений на С++: Пер. с англ. –М.: –СПб.: Издательство Бином, Невский диалект, 1999. –560 с.: ил.

[4] Zubeck J. Повторное использование объектов в системах укоренной разработки приложений // COMPUTERWEEKLY. – 1998. – №7.– C.24-28

[5] Трофимов С.А. CASE-технологии: практическая работа в Rational Rose. –M: ЗАО Издательство БИНОМ, 2001 г. -272 с.: ил.

[6] Брукс Ф. Мифический человеко-месяц или как создаются программные системы.- Пер. с англ. – СПб.: Символ-Плюс, 1999. – 304 с.: ил.

[7] Бюрер К. От ремесла к науке: поиск основных принципов разработки ПО. (http://interface.ru/rational/science.htm)