

# МОДИФИЦИРУЕМОСТЬ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ – ОСНОВА ДЛЯ УВЕЛИЧЕНИЯ ЕГО ЖИЗНЕННОГО ЦИКЛА

Г. П. Косинов, Н. В. Водополова

Учреждение образования «Гомельский государственный  
технический университет имени П. О. Сухого»,  
кафедра «Информатика»

Практически любой программный комплекс на этапе внедрения достаточно хорошо встречается специалистами предприятия. Неудобства этого этапа для пользователя достаточно очевидны: большой ручной ввод (заполнение справочников, начальных остатков и т. д.) и первоначальные проблемы с новой схемой ввода данных, получением отчетности и т. д. Преимущества же в том, что данный программный комплекс в точности соответствует требованиям специалистов предприятия и текущей экономической ситуации в стране. Поэтому данный комплекс приобретается и устанавливается на компьютерах заказчика. Слабое звено в данной цепи то, что пользователи (а точнее даже не они, а руководители предприятия, часто имеющие отдаленное представление о разработке программного обеспечения) не подозревают о статичности программного обеспечения и даже небольшие изменения могут вызвать отказ от его использования. Поэтому проблема увеличения жизненного цикла программного обеспечения становится с нашей точки зрения достаточно весомой.

Большой интерес представляет собой процесс адаптации программных комплексов без участия разработчика (или лиц, осуществляющих сопровождение) и такое свойство программ как модифицируемость.

Модифицируемость программного средства (*changeability*) – совокупность свойств программного средства, характеризующая усилия, необходимые для внесения в него изменений, связанных с устранением дефектов или приведением в соответствие с изменившейся средой функционирования [п. 5.2 Прил. 2 ГОСТ 28806–90].

С точки зрения функционирования большинства разработанных программ выделены следующие этапы:

- «Инициализация АРМ, поддержка баз данных». Основные глобальные переменные, необходимые для функционирования программного комплекса, должны выноситься во внешние текстовые файлы, легко доступные для редактирования, а поддержка баз данных осуществляться стандартными средствами используемой СУБД.

• «Организация пользовательского интерфейса». Система меню, контекстно-зависимой помощи, организации диалога также должна быть представлена внешними файлами, доступными для редактирования пользователю. Систему меню предлагается организовать в виде свободной таблицы базы данных, в которой будут представлены поля: ПолеРодитель, ПолеПотомок, ИмяВызываемойПроцедуры, а также поля с кодом горячей клавиши для вызова процедуры и координатами отображения меню (таблица).

#### Пример организации файла меню

ПолеРодитель	ПолеПотомок	ИмяВызываемойПроцедуры
ОсновноеМеню	Справочники	–
ОсновноеМеню	Расчет	Rascet()
ОсновноеМеню	Загрузка	–
Справочники	Банки	Banks()
Справочники	Предприятия	–
Справочники	Товары	Tovars()
Справочники	План счетов	Plans()
Предприятия	РБ	Pred1()
Предприятия	Иностранные	Pred(2)

При запуске программы происходит фильтрация таблицы по критерию ПолеРодитель=«ОсновноеМеню» и на экране отображаются только записи ПолеПотомок (в данном случае «Справочники», «Расчет», «Загрузка»). При выборе любой из записей меню (например, «Справочники»), старый фильтр снимается и создается новый по критерию ПолеРодитель=«Справочники» и опять отображаются только поля-потомки для родителя «Справочники» (в данном случае «Банки», «Предприятия», «Товары», «План счетов»). Если поле ИмяВызываемойПроцедуры не является пустым, то вместо отображения подменю, вызывается соответствующая процедура. Также необходимо предусмотреть возможность вызова из меню других исполняемых файлов, а также внешних файлов-сценариев.

• «Расчет». Самый сложный этап как для пользователя, так и для разработчика. Должен быть предусмотрен вынос расчетных алгоритмов во внешние файлы (файлы-сценарии), однако большие участки программы предпочтительнее держать в исполняемом коде из-за статичности учета и медленного анализа внешнего файла (если алгоритм большой и сложный). Также необходимо предусмотреть вынос во внешние файлы всевозможных расчетных коэффициентов, а также при необходимости и самих формул.

• «Организация выходных форм». Пользователю должны быть доступны для редактирования не только полученные в результате работы документы, но и модели их организации (сценарии печати). Также предпочтительно организовывать вывод данных с использованием макросов в *MS Excel* или *MS Word*.

Данный подход позволяет пользователю без участия разработчика влиять на функционирование всего программного комплекса в целом, а также при необходимости подвергать его модификации.