

**МИНИСТЕРСТВО ВЫСШЕГО И СРЕДНЕГО СПЕЦИАЛЬНОГО
ОБРАЗОВАНИЯ РЕСПУБЛИКИ УЗБЕКИСТАН**

**ТАШКЕНТСКИЙ ГОСУДАРСТВЕННЫЙ ЭКОНОМИЧЕСКИЙ
УНИВЕРСИТЕТ.**

**Факультет « Информационные технологии
и менеджмент».**

С.С. Гулямов, А.А. Мусалиев, Б.А. Бегалов

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ИНФОРМАЦИОННЫХ СИСТЕМ

УЧЕБНО – МЕТОДИЧЕСКОЕ ПОСОБИЕ

ТАШКЕНТ – 2006г.

Оглавление

Введение	3
1. Общая характеристика и классификация программного обеспечения.....	3
2. Общесистемное программное обеспечение.....	6
3. Прикладное программное обеспечение	8
4. Построение программного обеспечения информационных систем..	12
5. Жизненный цикл программного продукта, стадии разработки.....	13
6. Методы проектирования программных продуктов.....	21
7. Показатели качества программных продуктов.....	25
8. Методы оценки затрат на разработку программных средств.....	27
9. Экономическая эффективность программного изделия (продукта).	28
10. Сертификация программных продуктов.....	30
11. Выбор и оценка программных продуктов.....	31
12. Категория специалистов, связанных с созданием и эксплуатации программ.....	35
13. Правовые методы защиты программных продуктов.....	37
Кейс-стади. «Проектирование алгоритмов и программ».....	41
Использованная литература.....	47

Введение

Общепризнанно, что XXI век – это век информации и развития информационно – коммуникационных технологий: повсеместно появляются совершенно новые направления человеческой деятельности, связанных с обработкой, хранением и передачей информации, разработкой программной продукции, созданием и эксплуатацией информационных систем и т. д.

Создание и использование информационных систем и технологий для любой организации нацелены на решение следующих задач:

- структура информационных систем, их функциональное назначение и используемые информационные технологии должны соответствовать целям, стоящим перед организацией.
- производство достоверной, надежной, своевременной и систематизированной информации, как важнейшего ресурса в управленческих процессах.

Проектирование информационной системы, конечным продуктом которого является утвержденная проектно – техническая документация (ПТД) и программный продукт (ПП) на конкретную систему представляет собой сложный технологический процесс по принятию проектных решений. В процессе проектирования информационной системы возникает широкий спектр задач выбора и оценки проектных решений, связанных с программным обеспечением, в том числе:

- выбор версии операционной системы;
- выбор системы программирования;
- выбор системы управления базами данных;
- выбор пакетов прикладных программ (ППП);
- выбор инструментальных средств проектирования;
- выбор средств автоматизированного проектирования.

Процессы разработки информационных систем требуют больших материальных и трудовых затрат. Не менее значительными бывают затраты и при их полном или частичном приобретении. Некачественность всей информационной системы и информационных технологий или элементов приводит к значительным потерям, иногда к катастрофе.

1. Общая характеристика и классификация программного обеспечения

Общественное разделение труда привело к превращению сферы производства программных продуктов в составную часть общественного материального производства.

Под программным обеспечением понимается совокупность программ для реализации целей и задач информационной системы, а также нормального функционирования комплекса технических средств.

Программа (program) – упорядоченная последовательность команд (инструкций) компьютера для решения задачи.

Программное обеспечение (soft ware) – совокупность программ обработки данных и необходимых для их эксплуатации документов.

Программы предназначены для машинной реализации задач. Термины «задача» и «приложения» имеют очень широкое употребление в контексте программного обеспечения.

Задача (problem task) – проблема, подлежащая решению.

Приложение (application) – программная реализация на компьютере решения задачи.

С позиции специфики разработки и вида программного обеспечения различаются два класса задач: технологические и функциональные.

Технологические задачи решаются при организации технологического процесса обработки информации на компьютере и используются для обеспечения работоспособности компьютера, разработки других программ или обработки данных функциональных задач.

Функциональные задачи реализуют функции управления в рамках информационных систем предметных областей, т. е. реализуют цели и задачи информационной системы. Функциональные задачи в совокупности образуют предметную область и полностью определяет ее специфику.

Предметная (прикладная) область (application domain) – совокупность связанных между собой функций, задач управления, с помощью которых достигается выполнение поставленных целей.

По длительности жизненного цикла программное обеспечение можно разделить на два класса: с малым и большим временем жизни. Этим классам программ соответствует гибкий (мягкий) подход к их созданию и использованию, как к объектам научного творчества или произведениям «искусства» и жесткий промышленный подход, применяемый при регламентированном проектировании и эксплуатации промышленных изделий. В научных организациях и малых фирмах преобладают разработки программ первого класса, а в проектных и системных организациях – второго.

Программы с малой длительностью эксплуатации создаются в основном для решения научных и инженерных задач с целью получения конкретных результатов. Такие программы относительно невелики, разрабатываются одним специалистом или небольшой группой, обычно они не предназначены для тиражирования и передачи в другие коллективы.

Программы с большой длительностью эксплуатации создаются для регулярной обработки информации и управления в процессе функционирования сложных вычислительных систем. Размеры программ могут изменяться в широких пределах (тысячи – миллион команд), однако все они обладают свойством познаваемости и возможности модификации в процессе длительного сопровождения и использования различными специалистами. Программы данного класса допускают тиражирование, они

оформляются документацией как промышленное изделие и представляют собой отчужденный программный продукт (ПП).

Проектирование и эксплуатацию программ могут осуществлять большие коллективы специалистов. В связи с этим необходима формализация требуемых технологических характеристик комплексов программ и их компонентов, а также формализованные испытания и определение достигнутых показателей качества.

Программный продукт – комплекс взаимосвязанных программ для решения определенной проблемы (задачи) массового спроса, подготовленный к реализации как любой вид промышленной продукции.

Программные продукты создаются производителями чаще всего не для собственного потребления, а для продажи на внутреннем и внешнем рынках и, предназначены для удовлетворения потребностей пользователей по автоматизации процессов управления.

Программный продукт должен быть соответствующим образом подготовлен к эксплуатации, иметь необходимую техническую документацию, представлять сервис и гарантию надежной работы, иметь товарный знак изготовителя, а также желательно наличие кода государственной регистрации. Только при таких условиях программный комплекс может быть назван программным продуктом.

Программные продукты представляют собой специально упакованные и оформленные для коммерческой продажи, проката, сдачу в аренду, или лизинга пакеты программ, разработанные и/или поставляемые системными или независимыми поставщиками. Они не включают специально разработанные прикладные программные решения, которые фирмы-разработчики «под ключ» дополняют закупаемые им у фирм-производителей или у третьих фирм вычислительные системы.

Основные характеристики программного продукта. Основными характеристиками программного продукта являются:

- алгоритмическая сложность (логика алгоритмов обработки информации);
- состав и глубина проработки реализованных функций обработки;
- полнота и системность функций обработки;
- объем файлов программ;
- требования к операционной системе и техническим средствам обработки со стороны программного продукта;
- объем дисковой памяти;
- объем оперативной памяти для запуска программ;
- тип процессора;
- версия операционной системы;
- наличие вычислительной сети.

Структура программного обеспечения. Исходя из специфики информационных систем, в состав программного обеспечения входят

общесистемные и специальные (прикладные) программные средства, а также техническая документация (рис 1).

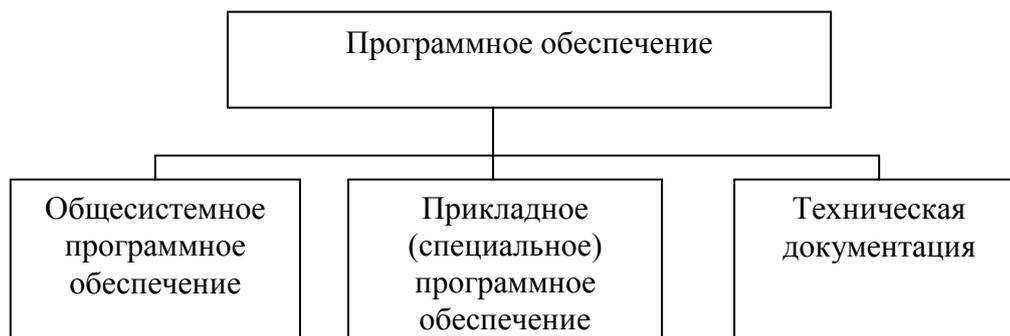


Рис 1. Структура программного обеспечения

К общесистемному программному обеспечению относятся программные средства, предназначенные для решения типовых задач обработки данных, а также для обеспечения работы компьютера и компьютерных сетей и разработки новых программ.

Специальное (прикладное) программное обеспечение представляет собой совокупность программ разработанных и/или использованных при создании конкретной информационной системы, реализующие его цели и задачи. В его состав входят программные средства, реализующие разработанные модели разной степени адекватности, отражающие функционирование реального объекта.

2. Общесистемное программное обеспечение

В общесистемное программное обеспечение входят программные средства предназначенные для обеспечения деятельности компьютерных систем как таковых. В их составе выделяют: базовое программное обеспечение, сервисное программное обеспечение, инструментарий технологии программирования (рис 2).



Рис 2. Структура общесистемного программного обеспечения

Базовое программное обеспечение (base software) – минимальный набор программных средств, обеспечивающих работу компьютера. В базовое программное обеспечение входят: операционная система, операционные оболочки, сетевая операционная система.

Операционная система предназначена для управления выполнением пользовательских программ, планирования и управления вычислительными ресурсами ЭВМ.

Сетевые операционные системы – комплекс программ, обеспечивающих обработку, передачу и хранение данных в сети. Сетевая операционная система предоставляет пользователям различные виды сетевых услуг (управление файлами, электронная почта, процессы управлению сетью и др.), поддерживают работу в абонентских системах. Сетевые операционные системы используют архитектуру клиент-сервер или одноранговую архитектуру. Сетевые операционные системы поддерживают не только локальные вычислительные сети, но и распространяются на ассоциации локальных сетей.

Сервисное программное обеспечение – программы и программные комплексы, которые расширяют возможности базового программного обеспечения. Расширением базового программного обеспечения компьютера является набор сервисных, дополнительно устанавливаемых программ, которые можно классифицировать по функциональному признаку следующим образом:

- программы диагностики работоспособности компьютера;
- антивирусные программы, обеспечивающие защиту компьютера, обслуживание и восстановление зараженных файлов;
- программы обслуживания дисков, обеспечивающие проверку качества поверхности магнитного диска, контроль сохранности файловой системы на логическом и физическом уровнях, сжатие дисков, создание страховых копий дисков, резервирование данных на внешних носителях и др.;
- программы архивирования данных, которые обеспечивают сжатие информации в файлах с целью уменьшения объема памяти для ее хранения;
- программы обслуживания сети.

Эти программы часто называют утилитами.

Инструментарии технологии программирования – *программотехника (software engineering)* – технология разработки, отладки, верификации и внедрения программного обеспечения. Инструментарии технологии программирования – программные продукты поддержки (обеспечения) технологии программирования.

В рамках этих направлений сформировались следующие группы программных продуктов: средства для создания приложений; средства для создания информационных систем

Средства для создания приложений включают: локальные средства, обеспечивающие выполнение отдельных работ по созданию программ;

интегрированные среды разработчиков программ, обеспечивающие выполнение комплекса взаимосвязанных работ по созданию программ.

Средства для создания приложений – это совокупность языков и систем программирования, а также различные программные комплексы для отладки и поддержки создаваемых программ.

Локальные средства разработки программ включают языки и системы программирования, а также инструментальную среду пользователя.

Интегрированные среды разработки программ объединяют набор средств для комплексного применения их на всех технологических этапах создания программ. Основное назначение инструментарии данного вида – повышение производительности труда программистов, автоматизация создания кодов программ, обеспечивающих интерфейс пользователя графического типа, разработка приложений для архитектуры клиент-сервер, запросов и отчетов.

Средства для создания информационных систем (CASE-технологии) – средства представляющие методы анализа, проектирования и создания программных систем и предназначенных для автоматизации процессов разработки и реализации информационных систем.

CASE-технология – программный комплекс, автоматизирующий весь технологический процесс анализа, проектирования, разработки и сопровождения сложных программных систем.

Средства CASE-технологии делятся на две группы:

- встроенные в систему реализации – все решения по проектированию и реализации привязаны к выбранной системе управления базами данных (СУБД);
- независимые от системы реализации – все решения по проектированию ориентированы на унификацию начальных этапов жизненного цикла и средств их документирования, обеспечивает большую гибкость в выборе средств реализации.

Основное достоинство CASE-технологий – поддержка коллективной работы над проектом за счет возможности работы в локальной сети разработчиков, экспорта/импорта любых фрагментов проекта, организационного управления проектом.

3. Прикладное программное обеспечение

Прикладное программное обеспечение используется и разрабатывается для решения конкретных задач пользователей и включает прикладные программы и пакеты прикладных программ (ППП).

Прикладное программное обеспечение подразделяются на:

- программы индивидуального пользования (они специально разработаны для конкретного пользователя или группы пользователей и не продаются другим лицам);
- программы общего назначения, разработанные для свободной продажи широкому кругу лиц.

При разработке программ общего назначения фирма-разработчик, с одной стороны, должна обеспечить универсальность выполняемых функций обработки данных, с другой стороны, гибкость и настраиваемость программного продукта на условия конкретного применения. Отличительной особенностью программного продукта общего назначения должна быть их системность – функциональная полнота и законченность реализуемых функций обработки данных, которые применяются в совокупности.

Значительное место в прикладном программном обеспечении занимают ППП, служащие программным инструментом решения функциональных задач и являющихся самым многочисленным классом программных продуктов. В данный класс входят программные продукты, выполняющие обработку информации различных предметных областей.

Пакет прикладных программ (application program package) – комплекс взаимосвязанных программ для решения задач определенного класса конкретной предметной области.

Пакеты прикладных программ являются мощным инструментом автоматизации решаемых пользователями задач, практически полностью освобождая его от необходимости знать, как компьютер выполняет те или иные процедуры преобразования информации. Именно ППП служат важнейшим источником развития программного обеспечения современных вычислительных систем, ускоряет и облегчает внедрение компьютерной техники в различные сферы деятельности.

Пакет прикладных программ (ППП) представляет собой комплекс программных средств и документов, предназначенных для реализации функционально завершенного алгоритма обработки данных и обеспечивающих автоматизацию разработки рабочих программ и автоматическое управление прохождением задач.

Программные средства пакета содержат:

- набор программных модулей, называемых телом пакета, из которых в соответствии с требованием пользователей по заданному алгоритму набирается конкретная рабочая программа;
- управляющую программу для управления и обслуживания пакета;
- средства генерации рабочих программ для конкретного применения, которые, принимая от пользователя информацию о требуемой модификации программ и ее составе, формируют из набора стандартных модулей законченные содержательные программы для реализации заданного алгоритма в конкретных условиях. Это процесс генерации и настройки пакета обеспечивает значительную его гибкость.

Кроме комплекса настраиваемых программ, пакет содержит документацию для разработки рабочих программ, с помощью которой пользователь может освоить этот пакет. Пакет должен отвечать определенным требованиям, предъявляемым к нему пользователем: широта применения, надежность и простота в эксплуатации, совместимость с программным обеспечением ЭВМ, наличие технической документации для его ввода в эксплуатацию.

Прикладные программы создаются пользователем с использованием инструментария технологии программирования. Создание и отладка программ осуществляется в соответствии с правилами и соглашениями того ППП, систем программирования, операционной системы и т. д., в рамках которого они применяются.

Классификация пакетов прикладных программ. Пакеты прикладных программ по сфере своей деятельности подразделяются на следующие классы:

Проблемно – ориентированные ППП направлены на определенный узкий круг задач и является самым представительным классом программных продуктов. Их классификация проводится по разным признакам:

- типам предметных областей;
- информационным системам;
- функциям и комплексам задач, реализуемым программным способом и др.

Для некоторых предметных областей возможна типизация функций управления, структуры данных и алгоритмов обработки, что позволяет разработать ППП одинакового функционального назначения. Таким образом, создается рынок программных продуктов:

- ППП автоматизированного бухгалтерского учета;
- ППП финансовой деятельности;
- ППП управления персоналом (кадровый учет);
- ППП управления материальными запасами;
- Финансовые информационные банковские системы и т. п.

Основными тенденциями в области развития проблемно – ориентированных программных средств являются:

- создание программных комплексов в виде автоматизированных рабочих мест (АРМ) управленческого персонала;
- создание интегрированных систем управления предметной областью на базе вычислительных сетей, объединяющих АРМы в единый программный комплекс с архитектурой клиент – сервер;
- организация данных больших информационных систем в виде распределенных баз данных на сети ЭВМ;
- наличие простых языковых средств конечного пользователя для запросов к базе данных;
- настройка функций обработки силами конечных пользователей (без участия программистов);
- защита программ и данных от несанкционированного доступа (парольная защита на уровне функции, режимов работы, данных).

Для подобного класса программ высоки требования к оперативности обработки данных, велики объемы хранимой информации, что обуславливает повышенные требования к средствам администрирования данных базы данных (актуализации, копирования, обеспечение производительности обработки данных).

Наиболее важно для данного класса программных средств создание дружественного интерфейса для конечных пользователей.

Данный класс программных продуктов весьма динамичен как по составу реализуемых ими функций, так и по используемому для их создания инструментарию разработчика.

Методо-ориентированные ППП. Данный класс ППП включает программные продукты, обеспечивающие независимо от предметной области и функций информационной системы математические, статистические и другие методы решения задач.

Наиболее распространены методы математического программирования, решения дифференциальных уравнений, имитационного моделирования, исследования операций.

На базе методов сетевого планирования с экономическими показателями проекта, формированием отчетов различного вида оформилось новое направление программных средств – *управление проектами*, пользователями этих программ являются менеджеры проектов. Эти программы предназначены для планирования и управления ресурсами различных видов (материальными, техническими, финансовыми, кадровыми, информационными) при реализации сложных научно-исследовательских и проектных работ.

Офисные ППП. Данный класс программных продуктов охватывают программы, обеспечивающие управление деятельностью офиса: оргнайзеры (планировщики); программы-переводчики, средства проверки орфографии и распознавание текста; коммуникационные ППП.

ППП автоматизированного проектирования. Программы этого класса предназначены для поддержки работы конструкторов и технологов, связанных с разработкой чертежей, схем, диаграмм, графическим моделированием и конструированием, созданием библиотеки стандартных элементов чертежей и их многократным использованием, созданием демонстрационных иллюстраций и мультфильмов.

ППП общего назначения. Данный класс ППП содержит программные продукты, поддерживающих преимущественно информационные технологии конечных пользователей. Кроме конечных пользователей, этими ППП за счет встроенных средств технологий программирования могут пользоваться и программисты для создания сложных программ обработки данных.

Представителями данного класса ППП являются: настольные системы управления базами данных, серверы баз данных, генераторы (серверы) отчетов, текстовые процессоры, табличные процессоры, средства презентационной графики, интегрированные пакеты.

Настольные издательские системы. Данный класс ППП предназначен для профессиональной издательской деятельности, позволяющей осуществлять электронную верстку широкого спектра основных типов документов типа: информационного бюллетеня, краткой цветной брошюры, объемного каталога, торговой заявки, справочника и т. д.

Программные средства мультимедиа. Этот класс программ предназначен для отображения и обработки аудио и видеoinформации. Помимо программных средств, компьютер при этом должен быть оборудован дополнительными платами, позволяющими осуществлять ввод-вывод аналоговой информации.

Среди мультимедиа программ можно выделить две больших группы: пакеты программ для обучения и досуга и пакеты программ для создания мультимедиа - представлений, демонстрационных дисков и стендовых материалов.

Системы искусственного интеллекта. Данный класс программных продуктов реализует отдельные функции интеллекта человека. Основными компонентами систем искусственного интеллекта являются базы знаний, интеллектуальный интерфейс с пользователем и программы формирования логических выводов.

4. Построение программного обеспечения информационных систем

Автоматизированные информационные системы (АИС) создаются для накопления, обработки и передачи информации и строятся на технических средствах и программном обеспечении (базовом и сервисном), системном и прикладном, которые доходят до конечного пользователя в виде *информационных технологий*. Без машинных и системных программ компьютер представляет не более чем металлическую коробку, а без прикладных программ невозможно существование автоматизированной информационной системы.

Прикладная программа реализует информационные процессы в конкретной предметной области и опирается на машинные и системные программы.

Создание автоматизированной информационной системы начинается с фундамента: приобретаются технические средства с соответствующими машинными программами. Приобретенные компьютеры комплектуются системными программами.

Заложив фундамент АИС, можно приступать к строительству «первого этажа» определить и применить типовые или разработать новые конкретные операции (функции) по обработке информации в системе. Набор таких типовых функций образуют функциональные или функционально-ориентированные технологии. Такие технологии универсальны и разрабатываются при минимальном участии будущего пользователя. Инструментальную основу таких технологий составляют системные программы.

Над первым этажом возводится «второй». На базе автономных типовых функций обработки информации создаются технологические подсистемы: информационно-поисковые системы, базы данных и базы знаний, экспертные системы, офисные системы, системы коллективного пользования и др., что позволит ускорить процесс ввода прикладных систем

в эксплуатацию. В АИС часто применяют уже отработанные типовые технологические решения, которые реализуют конкретные типовые прикладные проблемы и формируется разработчиками системы путем обобщения функциональных требований со стороны уже действующих прикладных информационных систем.

Далее, в «общем» каркасе АИС выделяются проблемные или проблемно-ориентированные информационные технологии. Потенциальные пользователи такой технологии могут принять участие в ее разработке только на начальной стадии формирования требований или типизации конкретных задач или на конечной стадии – при разработке специализированных дополнений. Это позволяет основную часть технологий создавать автономно от пользователя и применять унифицированные технические решения.

«Третий этаж» АИС строится из прикладных программ, а соответствующие информационные технологии определяются как предметные или предметно-ориентированные.

Они должны максимальным образом удовлетворять частным требованиям данного приложения и их появление невозможно без участия будущего пользователя. Таким образом, «третий этаж» возведен, и здание АИС построено. Его можно отдавать в эксплуатацию пользователю.

Выбор технических средств, а следовательно машинных программ, определяет выбор и системных программ, на которых реализуется сначала автономные типовые операции обработки информации, функционирующие в созданной системной среде, затем выбираются типовые или создаются заново необходимые технологические подсистемы, а уже потом разрабатываются прикладные программы.

5. Жизненный цикл программного продукта, стадии разработки

Как и любое изделие, программный продукт (ПП) имеет свой цикл жизни, то есть интервал времени от начального момента возникновения объективной необходимости в ПП до момента изъятия его из эксплуатации. Жизненный цикл (ЖЦ) ПП заканчивается в результате его морального, а не физического износа, т. е. морально устарело, если оно перестает удовлетворять актуальным требованиям, а дальнейшая его модификация не представляется возможной или невыгодна, что влечет за собой необходимость в разработке нового ПП.

В настоящее время существует значительное количество моделей жизненного цикла программного продукта. С практической точки зрения отсутствие в использовании единой модели объясняется тем, что жизненный цикл моделируется или порождается под воздействием определенной внешней среды, а не наоборот. К основным объектам, формирующим внешнюю среду, относят: организационно-технические условия разработки программного продукта; условия их сопровождения и эксплуатации; технические условия, структуру и режим использования; опыт и организацию коллектива разработчиков; класс решаемых задач с учетом их

типа и прикладной области, выделение ресурсов (временных, трудовых, финансовых, материальных).

Жизненный цикл программного продукта. Программы любого вида характеризуются жизненным циклом, состоящим из отдельных этапов:

- a) маркетинг рынка программных средств, спецификация требований к программному продукту;
- b) предпроектное обследование;
- c) разработка технического задания;
- d) выбор методологии разработки программного продукта, проектирование и программирование;
- e) отладка и тестирование программ;
- f) документирование программного продукта, подготовка эксплуатационной и технологической документации;
- g) внедрение программного продукта;
- h) обучение заказчика.
- k) выход на рынок программных средств, распространение программного продукта;
- p) эксплуатация пользователем программного продукта;
- q) снятие программного продукта с продажи, отказ от сопровождения.

На рис 3. изображены этапы жизненного цикла программного продукта и показано их временное соответствие друг другу.

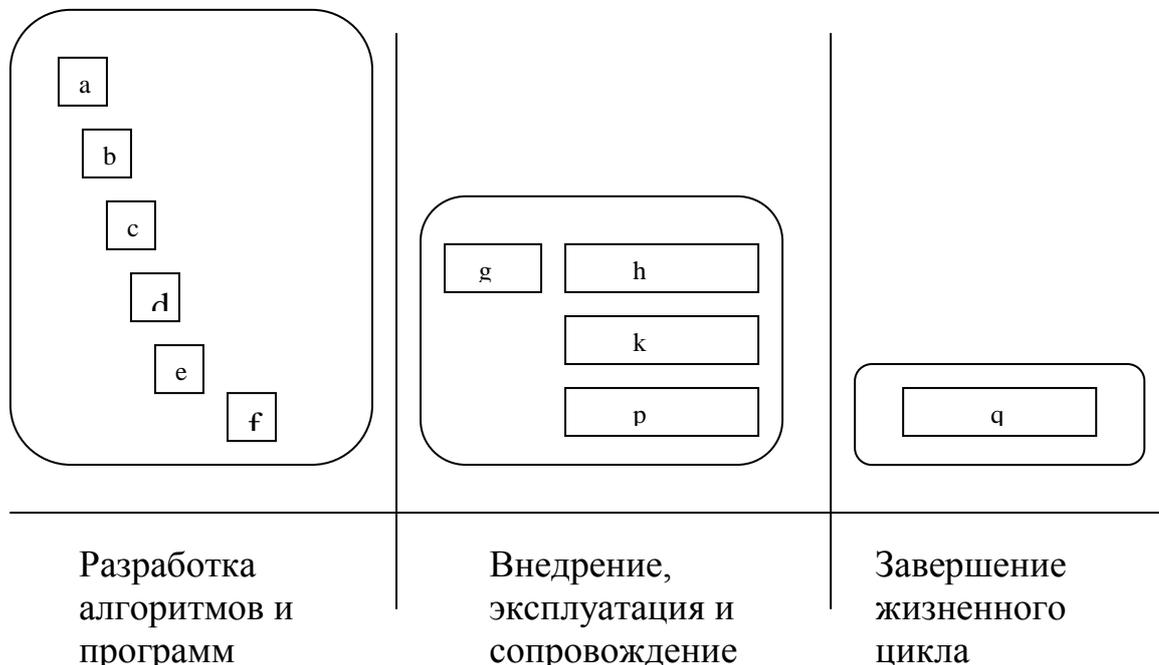


Рис 3. Этапы жизненного цикла программных продуктов

Ниже рассматривается содержание отдельных этапов жизненного цикла программных продуктов.

Этап маркетинга и спецификации программного продукта предназначены для изучения требований к создаваемому программному продукту, которое заключается в следующем:

- определение состава и назначения функции обработки данных программного продукта;
- установление требований пользователя к характеру взаимодействия с программным продуктом, типу пользовательского интерфейса (система меню, использование манипулятора – мышь, типы подсказок, виды экранных документов);
- требования к комплексу технических и программных средств для эксплуатации программного продукта и т. д.

Исходным этапом создания ПП является разработка требований, в процессе которого проводятся поисковые и исследовательские работы, формируется комплекс требований, выражающий потребности пользователя в конкретном ПП. Будущий комплекс программ тщательно анализируется с учетом выполняемых ими функций и основных свойств, обосновывается целесообразность их разработки, предварительно оцениваются трудовые и стоимостные затраты и сроки создания, вырабатываются рекомендации по выбору инструментальных средств и методов, которые предполагается использовать в процессе разработки программ. Обязательным в содержании данного этапа является также формирование требований к качеству программ в соответствии с условиями их функционирования и реализации конкретных функций. Выполнение этих работ позволяют предотвратить дополнительные расходы, вызванные модификацией программ при их внедрении и сопровождении.

Этот этап самый важный, так как неправильное формулирование требований приводит к выполнению ненужной работы, а недооценка сложности вызывает перерасход средств и времени. Так как требования к проекту во время работы над ними могут уточняться и меняться, а выполнение требований надо отслеживать, применяются специальные программы для управления требованиями.

Часто заказчик не в состоянии выразить, что он хочет и задача специалиста по системному анализу – помочь заказчику выразить свои требования в виде, пригодном для формализации. После согласования требований подписывается контракт на разработку ПП. В дальнейшем любые отклонения от сформулированных требований к изделию (как со стороны заказчика, так и со стороны исполнителя) рассматривается как нарушение контракта. Каждый этап требует от заказчика вложения собственных трудозатрат и привлечения высокопрофессиональных специалистов, поэтому он обязательно должен оплачиваться. Однако, хотя первый этап очень важен, заказчик редко понимает эту важность и не готов платить достаточно большой суммы.

Примерный объем работ на этом этапе - 5% объема всего проекта.

Если программный продукт создается не под заказ и предполагается выход на рынок программных средств, маркетинг выполняется в полном

объеме: изучаются программные продукты – конкуренты и аналоги, обобщаются требования пользователей к программному продукту, устанавливается потенциальная емкость рынка сбыта, дается прогноз цены и объема продаж. Кроме того, важно оценить необходимые для разработки программного продукта материальные, трудовые и финансовые ресурсы, ориентировочные длительности основных этапов жизненного цикла программного продукта.

Если программный продукт создается как заказное программное изделие для определенного заказчика, также важно сформулировать и документировать задание на его разработку, что выполняется на этапе предпроектного обследования и разработки технического задания.

Этап предпроектного обследования. На этом этапе осуществляется обследование объекта автоматизации. С помощью инструментальных средств составляется формальная модель его работы, модель базы данных, объектов и потоков информации. На этом этапе привлекаются специалисты заказчика и эксперты, хорошо знакомые с предметной областью, для которого составляется задача.

Примерный объем работ второго этапа – 10% от общего объема работ.

Этап разработки технического задания. На основе формальной модели составляется подробное техническое задание для программистов, спецификация отдельных модулей, таблицы баз данных, другая сопроводительная документация. Готовится подробный календарный план работ.

Примерный объем этого этапа работ – 10% от общего объема работ.

Этапы выбора методологии разработки ПП, проектирования и программирования.

Проектирование программ связано с алгоритмизацией процесса обработки данных, детализацией функций обработки разработкой структуры программного продукта (архитектуры программных модулей), структуры информационной базы задачи, выбором методов и средств создания программ – технологии программирования. Программирование является технической реализацией проектных решений и выполняется с помощью выбранного инструментария разработчика.

Выбор методологии разработки ПП. Нужно отметить, что крупные компании имеют собственные методологии, ориентированные на конкретные задачи, однако многие методологии имеют общие черты. Существуют методологии структурного (нисходящего) и объектного проектирования. Достаточно популярна методология итерационного проектирования, ориентированная на использовании RAD средств и систем автоматической генерации исходных текстов на основе созданной исходной модели. Такой подход хорош тем, что позволяет быстро создать первый работающий прототип программы, когда требования к ней окончательно не определены, а в дальнейшем, на следующих итерациях детализировать постепенно и реализовать конкретные возможности, пропущенные по каким-либо причинам на предыдущей итерации.

Проектирование является следующим этапом в процессе создания ПП, в процессе которого требования пользователей формулируются в более точном и конкретном виде. Проектирование программ охватывает комплекс работ по разработке структуры программ и их компонентов; выбору языка программирования и конкретной конфигурации комплекса технических средств, на котором предполагается реализация разрабатываемых программ. В процессе проектирования решается задача выбора оптимальной структуры программ, определяющая содержание и характер работы на последующих этапах разработки ПП. На данном этапе качество ПП обеспечивается конкретными решениями и зависит в основном от организации управления разработкой, квалификации специалистов, использование прогрессивных методов, приемов, правил и средств проектирования программ.

После проектирования программ следует их кодирование. На практике эти этапы часто перекрываются, т. е. за проектированием отдельных модулей выполняется их программирование, а затем, возможно, и предварительная проверка правильности функционирования разработанного модуля.

Программирование характеризуется большим числом разнообразных правил, приемов, методов и средств их выполнения, применение которых зависит от квалификации, опыта и индивидуальных особенностей программистов. Этот этап наиболее автоматизирован в процессе разработки ПП. В настоящее время существует множество языков программирования и средств автоматизации, облегчающих труд программистов и повышающие его производительность, а также согласующие предпосылки унификации и стандартизации процесса создания ПП. К тому же использование современных приемов программирования, средств автоматизации, проведение различного рода проверок и контроля программирования, способствуют предотвращению и выявлению значительного числа ошибок, что сокращает время и расходы на этапе отладки и тестирования программ, повышает их качество.

Для больших и сложных программных комплексов, имеющих развитую модульную структуру построения, отдельные работы по проектированию и программированию могут выполняться параллельно, обеспечивая сокращение общего времени разработки программного продукта.

Примерный объем этапов работ выбора методологии разработки ПП, проектирования и программирования – 10% от общего объема работ.

Этап отладки и тестирования программ, следующий после программирования, имеет целью выявление и устранение ошибок в них, а также ее определение, в какой мере разработанная программа удовлетворяет требованиям, сформулированным в спецификации. Работы по отладке и тестированию программ характеризуются большой степенью повторяемости, и является наиболее утомительным и дорогостоящим. В связи с этим уделяется большое внимание разработке и использованию различных системных и инструментальных средств, автоматизирующих выполнение работ на данном этапе, что позволяет повысить качество разрабатываемых программ и снизить трудоемкость их создания.

Тестирование программ происходит в два этапа: проверка ее на наличие ошибок у разработчиков программы и после анализа и устранения ошибок – у заказчика.

Примерный объем работ – 10% от общего объема проекта.

Документирование программного продукта является обязательным видом работ и должна содержать необходимые сведения по установке и обеспечению надежной работы программного продукта, поддерживать пользователей при выполнении функции обработки, определять порядок комплексирования программного продукта с другими программами. Успех распространения и эксплуатации программного продукта в значительной степени зависит от качества его документации.

На машинном уровне программного продукта, как правило, создаются:

- автоматизированная контекстно-зависимая помощь (HELP);
- демонстрационные версии, работающие в активном режиме по типу обучающих систем (электронный учебник) или пассивном режиме (ролик, мультфильм) для демонстрации функциональных возможностей программного продукта и информационной технологии его использования.

Этап внедрения ПП. После того, как заказчик удовлетворен качеством продукта, начинается его внедрение – подготовка к окончательному запуску в эксплуатацию. Проводятся работы по подготовке объекта к внедрению. Этот этап самый трудоемкий и «нудный» и занимает порой до 90% времени всего проекта.

Этап обучения заказчика. После того, как новая система готова к работе, сотрудников организации-заказчика нужно обучить работе с этой системой.

Примерный объем трудозатрат на обучение – 5% от общего объема проекта.

Выход программного продукта на рынок связан с организацией продаж массовому пользователю. Этот этап по возможности должен быть коротким, для продвижения программных продуктов применяются стандартные приемы маркетинга: реклама, увеличение каналов реализации, создание дилерской и дистрибьютерной сети, ценовая политика – продажа со скидками, сервисное обслуживание и др.

Требуется постоянная программа маркетинговых мероприятий и поддержки программных продуктов. Как правило, для каждого программного продукта существует своя форма кривой продаж, которая отражает спрос.

Этап эксплуатации программного продукта идет параллельно с его сопровождением, при этом эксплуатация программ может начинаться и в случае отсутствия сопровождения. Обычно, работы по сопровождению программного продукта (авторский надзор) начинается после того, как заказчиком ПП принят в эксплуатацию и продолжается определенный срок (обычно год). Это связано с тем, что будут возникать множество вопросов по работе в системе (возникновение ошибок, которые требуется устранить). Могут быть выпущены новые версии ПП и т. д.

Снятие программного продукта с продажи и отказ от его сопровождения происходит, как правило, в случае изменения технической политики фирмы – разработчика, неэффективной работы программного продукта, наличия в нем неустраняемых ошибок, отсутствие спроса.

Длительность жизненного цикла для различных программных продуктов неодинакова. Для большинства современных программных продуктов длительность жизненного цикла измеряется в годах (2-3 года). Хотя достаточно часто встречаются на компьютерах и давно снятие с производства программные продукты.

Особенность разработки программного продукта заключается в том, что на начальных стадиях принимаются решения, реализуемые на последующих этапах. Допущенные ошибки, например, при спецификации требований к программному продукту, приводят к огромным потерям на последующих этапах разработки или при эксплуатации программного продукта и даже к неудаче всего проекта.

Разработка программного продукта – это особый вид трудовой деятельности. Она характеризуется высокой динамичностью и значительной долей неопределенности получения результатов в установленные сроки. Этапы имеют размытые сроки начала и завершения, сложно переплетаются. По созданию форм и результатов труда процесс разработки программ близок к научно-исследовательским и опытно-конструкторским работам.

Стадии создания программных продуктов. При традиционной (неавтоматизированной) разработке программ независимо от принятого метода проектирования и используемого инструментария выполняются следующие работы: составление технического задания на программирование; технический проект; рабочая документация (рабочий проект) ввод в действие.

Составление технического задания на программирование. Данная работа соответствует этапу анализа и спецификации программ жизненного цикла программных продуктов.

При составлении технического задания требуется:

- определить платформу разрабатываемой программы (например, MS Windows, Unix, OS/2);
- определить необходимость сетевого варианта работы программы – определяется программное обеспечение вычислительной сети, допустимая номенклатура программного обеспечения сетевой обработки;
- определить необходимость разработки программы, которую можно переносить на различные платформы;
- обосновывать целесообразность работы с базами данных под управлением СУБД.

На этой же стадии выбирают методы решения задач, разрабатывают обобщенный алгоритм решения комплекса задач, функциональную структуру алгоритма или состав объектов, определяют требования к комплексу технических средств обработки информации, интерфейсу конечного пользователя.

Технический проект. На данной стадии выполняется комплекс наиболее важных работ:

- с учетом принятого подхода к проектированию программного продукта разрабатывается детальный алгоритм обработки данных или уточняется состав объектов и их свойств, методов обработки, событий, запускающих методы обработки;

- определяется состав общесистемного программного обеспечения, включающий базовые средства (операционную систему, модель СУБД, электронные таблицы, методо-ориентированные и функциональные ППП и т. п.);

- разрабатывается внутренняя структура программного продукта, образованная отдельными программными модулями;

- осуществляется выбор инструментальных средств разработки программных модулей.

Работы данного этапа в существенной степени зависят от принятых решений по технической части системы обработки данных и операционной среде, от выбранных инструментальных средств проектирования алгоритмов и программ, технологии работ.

Рабочая документация (рабочий проект). На данной стадии осуществляется базовых средств программного обеспечения (операционной системы, СУБД, методо-ориентированных ППП, инструментальных средств конечного пользователя – текстовых редакторов, электронных таблиц и т. п.). Выполняется разработка программных модулей или методов обработки объектов – собственно программирование или создание программного кода. Проводится автономная и комплексная отладка программного продукта, испытание работоспособности программных модулей и базовых программных средств. Для комплексной отладки готовится контрольный пример, который позволяет проверить соответствие возможностей программного продукта заданным спецификациям.

Основной результат работ этой стадии – также создание эксплуатационной документации на программный продукт:

- описание применения – дает общую характеристику программного продукта с указанием сферы его применения, требований к базовому программному обеспечению, комплексу технических средств;

- руководство пользователя – включает детальное описание функциональных возможностей и технологии работы с программным продуктом. Данный вид документации ориентирован на конечного пользователя и содержит необходимую информацию для самостоятельного освоения и нормальной работы пользователя (с учетом требуемой квалификации пользователя).

- руководство программиста (оператора) – указывает особенности установки (инсталляции) программного продукта и его внутренней структуры – состав и назначение модулей, правила эксплуатации и обеспечения надежной и качественной работы программного продукта.

В ряде случаев на данной стадии для программных продуктов массового применения создаются обучающие системы, демоверсии, гипертекстовые системы помощи.

Ввод в эксплуатацию. Готовый программный продукт сначала проходит опытную эксплуатацию (пробный рынок продаж), а затем сдается в промышленную эксплуатацию (тиражирование и распространение программного продукта).

6. Методы проектирования программных продуктов

Проектирование алгоритмов и программ – наиболее ответственный этап жизненного цикла программных продуктов, насколько создаваемая программа соответствует спецификациям и требованиям со стороны пользователей. Затраты на создание, сопровождение и эксплуатацию программных продуктов, научно-технический уровень разработки, время морального устаревания и много другое – все это также зависит от проектных решений.

Методы проектирования алгоритмов и программ очень разнообразны, их можно классифицировать по различным признакам, важнейшими из которых являются:

- степень автоматизации проектных работ;
- принятая методология разработки.

По степени автоматизации проектирования алгоритмов и программ можно выделить:

- методы традиционного (неавтоматизированного) проектирования;
- методы автоматизированного проектирования (CASE-технология и ее элементы).

Неавтоматизированное проектирование алгоритмов и программ преимущественно используется при разработке небольших по трудоемкости и структурной сложности программных продуктов, не требующих участия большого числа разработчиков. Трудоемкость разрабатываемых программных продуктов, как правило, небольшая, а сами программные продукты имеют прикладной характер.

Автоматизированное проектирование алгоритмов и программ возникло с необходимостью уменьшить затраты на проектные работы, сократить сроки их выполнения, создавать типовые «заготовки» алгоритмов и программ, многократно тиражируемых для различных разработок, координации работ большого числа разработчиков, стандартизации алгоритмов и программ.

Автоматизированное проектирование может охватывать все или отдельные этапы жизненного цикла программного продукта, при этом работы этапов могут быть изолированы друг от друга, либо составлять единый комплекс, выполняемых последовательно во времени. Как правило, автоматизированный подход требует технического и программного «перевооружения» труда самых разработчиков (мощных компьютеров,

дорогостоящего программного инструментария, а также повышения квалификации разработчиков и т. п.).

Автоматизированное проектирование алгоритмов и программ под силу лишь крупным фирмам, специализирующимся на разработке определенного класса программных продуктов, занимающих устойчивое положение на рынке программных средств.

Проектирование алгоритмов и программ может основываться на различных подходах, среды которых наиболее распространены:

- структурное проектирование программных продуктов;
- информационное моделирование предметной области и связанных с ней приложений;
- объектно-ориентированное проектирование программных продуктов.

Структурное проектирование. В основе структурного проектирования лежит последовательная декомпозиция, целенаправленное структурирование на отдельные составляющие.

Методы структурного проектирования представляет собой комплекс технических и организационных принципов системного проектирования.

Типичными методами структурного проектирования являются:

- нисходящее проектирование, кодирование и тестирование программ;
- модульное программирование;
- структурное программирование и др.

В зависимости от объекта структурирования различают:

- функционально-ориентированные методы – последовательное разложение задачи или целостные проблемы на отдельные, достаточно простые составляющие, обладающие функциональной определенностью;
- методы структурирования данных.

Для *функционально-ориентированных методов* в первую очередь учитываются заданные функции обработки данных, в соответствии с которым определяется состав и логика работы (алгоритмы) отдельных компонентов программного продукта. С изменением содержания функции обработки, их состава, соответствующего им информационного входа и выхода требуется перепроектирование программного продукта. Основной подход в структурном подходе делается на моделировании процессов обработки данных.

Для *методов структурирования данных* осуществляется анализ, структурирование и создание моделей данных, применительно к которым устанавливается необходимый состав функций и процедур обработки. Программные продукты тесно связаны со структурой обрабатываемых данных, изменения которых отражается на логике обработки (алгоритмах) и обязательно требует перепроектирование программного продукта.

При структурном подходе используют:

- диаграммы потоков данных (информационно-технологические схемы) – показывает процессы и информационные потоки между ними с учетом «событий», инициирующих процессы обработки;
- интегрированную структуру данных предметной области (инфологическая модель, ER-диаграммы);
- диаграммы декомпозиции – структуру и декомпозицию целей, функций управления, приложений;
- структурные схемы – архитектуру программного продукта в виде иерархии взаимосвязанных программных модулей с идентификацией связей между ними, детальная логика обработки данных программных модулей (блок-схемы).

Для полного представления о программном продукте необходима также текстовая информация описательного характера.

Информационное моделирование предметной области и связанных с ней приложений. Данный подход появился в условиях развития программных средств организации хранения и обработки данных – систем управления базами данных (СУБД). Выделяются следующие составляющие данного подхода:

- информационный анализ предметных областей (бизнес-областей);
- информационное моделирование – построение комплекса взаимосвязанных моделей данных;
- системное проектирование функций обработки данных;
- детальное проектирование процедур обработки данных.

Первоначально строятся информационные модели различных уровней представления:

- информационно-логическая модель, независимая от средств программной реализации хранения и обработки данных, отражающих интегрированные структуры данных предметной области;
- датологические модели, ориентированные на среду хранения и обработки данных.

Датологические модели имеют логический и физический уровни представления. Физический уровень соответствует организации хранения данных в памяти компьютера. Логический уровень соответствует хранению данных применительно к СУБД и реализуется в виде:

- концептуальной модели базы данных – интегрированные структуры данных под управлением СУБД;
- внешних моделей данных – подмножеств структур данных для реализации приложений.

Средствами структур данных моделируются функции предметной области, прослеживается взаимосвязь функций обработки, уточняется состав входной и выходной информации, логика преобразования входных структур данных в выходные. Алгоритмы обработки данных можно представить как совокупность процедур преобразования структур данных в соответствии с внешними моделями данных. Выбор средств реализации баз данных

определяет вид датологических моделей и, следовательно, алгоритм преобразования данных. В большинстве случаев используются реляционное представление базы данных и соответствующие реляционные языки для программирования (манипулирования) обработки данных СУБД и реализации алгоритмов обработки. Данный подход использован во многих CASE-технологиях.

Объектно-ориентированный подход. Объектно-ориентированный подход к проектированию программных продуктов основан на:

- выделения классов объектов;
- установления характерных свойств объектов и методов их обработки;
- создания иерархии классов, наследования свойств объектов и методов их обработки.

Каждый объект объединяет как данные, так и программу обработки этих данных и относится к определенному классу. С помощью класс один и тот же программный код может быть использован для относящихся к нему различных объектов.

Объектный подход при разработке алгоритмов и программ предполагает:

- объектно-ориентированный анализ предметной области;
- объектно-ориентированное проектирование.

Объектно-ориентированный анализ - анализ предметной области и выделение объектов, определение свойств и методов обработки объектов, установление взаимосвязей.

Объектно-ориентированное проектирование соединяет процесс объектной декомпозиции и представления с использованием моделей данных проектируемой системы на логическом и физическом уровнях, в статике и динамике.

Для проектирования программных продуктов разработаны объектно-ориентированные технологии, которые включают в себя специализированные языки программирования и инструментальные средства разработки пользовательского интерфейса.

Традиционные подходы к разработке программных продуктов всегда подчеркивали различие между данными и процессами их обработки. Так, технологии ориентированные на информационное моделирование, сначала специфицирует данные, а затем описывают эти процессы, использующие эти данные. Технологии структурного подхода ориентированы, в первую очередь, на процессы обработки данных с последующим установлением необходимого для этого данных и организации информационных потоков между связанными процессами.

Объектно-ориентированная технология разработки программных продуктов объединяют данные и процессы в логические сущности – объекты, которые имеют способность наследовать характеристики (методы и данные) одного и более объектов, обеспечивая тем самым повторное использование

программного кода. Это приводит к значительному уменьшению затрат на создание программных продуктов, повышает эффективность жизненного цикла программных продуктов (сокращается длительность фазы разработки).

7. Показатели качества программных продуктов

Качество программ в значительной степени определяет эффективность применения программ.

Программные продукты имеют многообразие показателей качества, которые отражают следующие аспекты:

- насколько хорошо (просто, надежно, эффективно) можно использовать программный продукт в его исходном виде;
- насколько легко эксплуатировать программный продукт (для понимания, модифицирования, повторных испытаний);
- можно ли использовать программный продукт при изменении условий его применения.

Дерево характеристик качества программных продуктов представлено на рис 4.

Каждая из характеристик нижнего уровня может быть разбита на более конкретные свойства, которые раскрывают ее содержание.

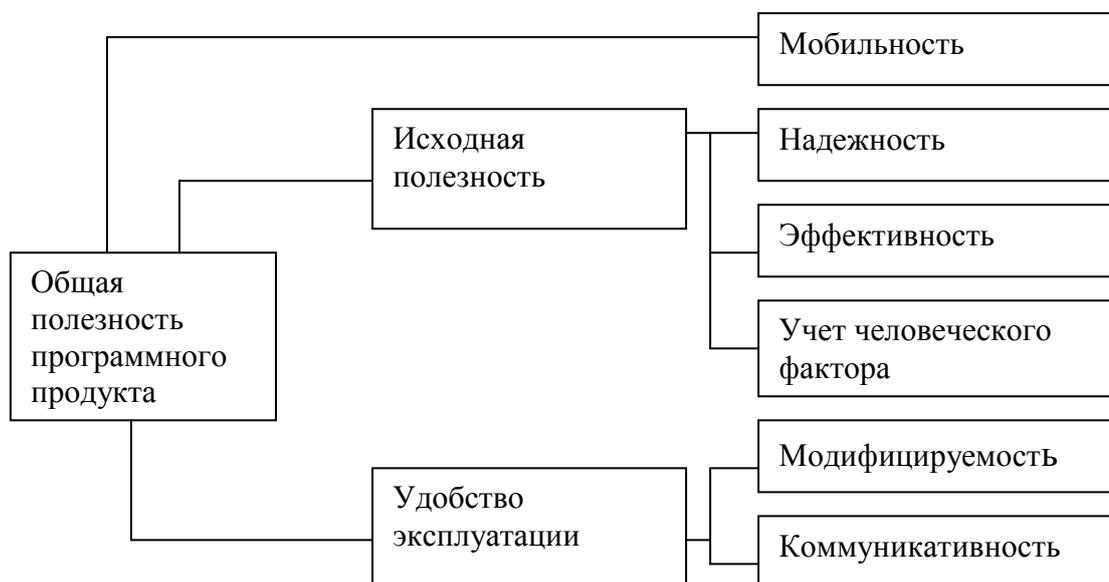


Рис 4. Дерево характеристик качества программного продукта

Ниже рассматривается совокупность характеристик качества программного продукта, количественная оценка которых позволяет определить, обладает ли данный программный продукт тем или иным свойством.

Мобильность программных продуктов означает их независимость от технического комплекса системы обработки данных (машиннезависимость), операционной среды, сетевой технологии обработки данных, специфики

предметной области и т. п. Мобильный (многоплатформенный) программный продукт может быть установлен на различных моделях компьютерах и операционных систем, без ограничений на его эксплуатацию в условиях вычислительной сети. Функции обработки такого программного продукта пригодны для массового использования без каких-либо изменений.

Надежность программного продукта определяется безбойностью и устойчивостью в работе программ, точностью выполнения предписаний функций обработки данных, возможности диагностики ошибок, возникающих в процессе работы программы.

Эффективность программного продукта заключается в выполнении требуемых функций при минимальных затратах ресурсов. Оценивается как с позиции его прямого назначения – требований пользователя, так и с точки зрения расходов вычислительных ресурсов, необходимых для его эксплуатации. Причем, под ресурсами подразумевается объем оперативной памяти, время работы процессора, объем внешней памяти, пропускная способность канала.

Учет человеческого фактора обозначает обеспечение дружественного интерфейса для работы конечного пользователя, наличие контекстно-зависимой подсказки или обучающей системы в составе программного обеспечения, хорошей документации для освоения и использования заложенном в программном средстве функциональных возможностей, анализ и диагностику возникающих ошибок и т. д.

Коммуникативность программных продуктов основано на максимально возможной их интеграции с другими программами, обеспечение обмена данными в общих форматах представления (экспорт \ импорт баз данных, внедрение или связывание объектов обработки и т. д.).

Качество программного продукта невозможно оценить с помощью единственного критерия качества. Оценка качества осуществляется рассмотрением множества критериев (показателей).

Необходимо отметить, что в каждом конкретном случае при оценке качества программного продукта пользователь должен подбирать определенный набор характеристик, удовлетворяющих его требованиям. После этого производится определение значений конкретных показателей. На основе найденных значений определяется некоторый интегральный показатель, который позволяет оценить программный продукт.

При анализе качества программных средств, прежде всего, оценивается пригодность программного средства для пользователя и удобство обслуживания для программиста. В простейшем случае, оценка качества осуществляется с использованием операционных показателей и технических показателей.

Операционные показатели характеризуют приспособленность программного средства к выполнению возложенных на него функций. К операционным показателям относятся такие показатели, как: простота общения пользователя с программным продуктом; удобства обучения и средства помощи; приспособленность к сопровождению.

Технические показатели программной реализации характеризуют эффективность программной реализации алгоритмов с точки зрения затрат памяти и машинного времени.

В условиях существования рынка программных продуктов, важными характеристиками программных продуктов являются также:

- стоимость;
- количество продаж;
- время нахождения на рынке (длительность продаж);
- известность фирмы разработчика и программы;
- наличие программных продуктов аналогичного назначения.

8. Методы оценки затрат на разработку программных средств

При определении трудовых и стоимостных затрат на разработку программных средств можно выделить четыре базовых направлений в развитии методов оценки:

1. Методы аналогий.
2. Нормативные методы.
3. Метод экспертных оценок.
4. Исследовательские методы.

Методы аналогии основаны на использовании в качестве базы для оценки трудовых и стоимостных затрат параметров ранее выполненных аналогичных программных разработок.

В этом случае предполагаемые затраты определяются по фактическим данным о расходах соответствующих программ аналогов с их корректировкой поправочными коэффициентами. Значения этих коэффициентов, а также выбор и сопоставление с аналогичными предшествующими разработками определяется экспертным путем.

Нормативные методы основаны на предварительном установлении нормативов затрат путем обобщения опыта создания конкретных программных средств. Дифференцированные нормативы затрат определяют величину ожидаемых затрат на разработку программных средств в зависимости от основных параметров. К числу основных параметров, влияющих на трудоемкость разработки, могут быть отнесены:

- объем программных средств в тысячах условных машинных команд;
- сложность разрабатываемых программных средств;
- степень новизны разрабатываемых программных средств;
- степень использования в разработке стандартных модулей, типовых программ и программных средств.

Использование нормативных методов расчетов затрат создают предпосылки для обоснованного планирования, учета и прогнозирования стоимостных и трудовых затрат на создание программных средств, их рационального использования. В то же время этот метод, представляет собой довольно общую оценку затрат. В них в должной мере не учитываются

организационно-технические факторы (методы выполнения и организация работ, используемые инструментальные средства, технологии разработки).

Учет организационно-технических факторов присущ методам экспертных и исследовательских оценок.

Методы экспертных оценок затрат на разработку программных средств представляют собой измерения, выполненный специалистом, имеющим должный опыт, знание, интуицию. Подобные методы не требуют больших затрат на получение оценок и сводится к формированию группы компетентных экспертов. Этот метод оценок оправдывается при определении затрат на уникальные программные разработки, отличающиеся новизной и не имеющих аналогов. Чаще же методы экспертных оценок используются в сочетании с другими методами.

При *исследовательском методе оценка затрат* на разработку программных средств является результатом специально проводимых исследований, экспериментов и наблюдений с последующей их статистической обработкой.

Следует различать общественно-необходимые затраты труда (ОНЗТ) и фактические затраты на производство конкретного программного продукта. Общественно-необходимые затраты труда базируются на прогрессивной технологии и эффективной организацией труда. Фактические затраты на разработку программного изделия определяется фактическим состоянием дел у конкретного производителя программного обеспечения. При ненаучной организации труда фактические затраты могут значительно превышать ОНЗТ.

Затраты в течение жизненного цикла программного продукта могут распределяться различными вариантами. При одном варианте минимизируются затраты на разработку, но при этом резко возрастают затраты на сопровождение. При другом варианте особое внимание уделяется возможности эффективности сопровождения. В этом случае сократятся общие затраты, но возрастут затраты на этапе разработки.

9. Экономическая эффективность программного изделия (продукта)

Экономическая эффективность программного изделия определяется сопоставлением получаемого эффекта от применения программного средства с затратами на программное средство. Если программное средство изготовлено на продажу (программный продукт), то экономический эффект от его использования распределяется между покупателями (пользователем) и продавцом (разработчиком программного средства).

Оценка эффективности программного средства для продавца и покупателя осуществляется поразному.

Для пользователя (покупателя) типового программного средства прибыль определяется сопоставлением получаемого экономического эффекта

от применения программного средства с затратами на его покупку и освоение.

Для продавца (разработчика) прибыль определяется сопоставлением вырученной суммы от продажи всех экземпляров программного продукта с затратами на его разработку.

При расчете показателей экономической эффективности различают:

- во-первых, экономический эффект от применения рассматриваемого программного средства;
- во-вторых, экономический эффект от применения программ, выполненных на базе рассматриваемого программного средства;
- в-третьих, экономический эффект от оказания услуг по сопровождению рассматриваемого программного средства.

Применение программного изделия предполагает повышение производительности труда пользователя программного средства. В простейшем случае, это означает, что пользователь выполняет тот же объем работ с меньшими затратами. Сопоставив затраты при первоначальном варианте и в случае использования рассматриваемого программного продукта, получаем оценку экономической эффективности.

Зависимость для определения экономического эффекта может быть представлена в следующем виде:

$$E = C_6 - C_n$$

где E – экономический эффект от внедрения новой разработки; C_6 – стоимость обработки экономической информации при использовании базового варианта; C_n – стоимость обработки экономической информации при предлагаемом варианте.

В том случае, если применяемое программное средство является составной частью принципиально новой системы, то расчет выполняется косвенным образом. Использование принципиально новой системы, как правило, приводит к созданию принципиально новой технологии работы пользователя. Это означает, что пользователь не будет выполнять тот же объем работы с меньшими затратами. Он будет выполнять совершенно новую работу. В этом случае оценивается эффект от использования всей системы. Эффект от использования конкретного программного продукта, являющегося составной частью системы, можно приближенно оценить пропорционально его доли в общем объеме программного обеспечения.

В качестве показателей экономической эффективности, как правило, используются показатели: годовая экономия; годовой экономический эффект; срок окупаемости.

Годовая экономия E_r , получаемая предприятием от применения данного программного средства, рассчитывается по формуле:

$$E_r = \sum_{k=1}^N V_k * (C_b - C_o),$$

где N – количество задач предприятия, разработанных с помощью данного программного средства; V_k – объем программного обеспечения по k – й задаче, выполненного с помощью данного программного средства; C_6 – стоимость разработки одной программной единицы на основе внедренного программного обеспечения.

Годовой экономический эффект F_r от внедрения рассматриваемого программного средства определяется по следующей формуле:

$$F_r = E_r - C_p,$$

где C_p – стоимость поставки рассматриваемого программного обеспечения.

Срок окупаемости определяет период, для которого экономия, получаемая от применения программного средства, равна затратам на его приобретение:

$$P = \frac{C_p}{E_r}$$

где P – срок окупаемости программного средства.

10. Сертификация программных продуктов

Сертификации программных продуктов в настоящее время во всем мире уделяется огромное внимание. Разработаны специальные методики и средства, стандарты по организации работы центров сертификации, стандарты на показатели качества программного обеспечения, стандартизация процедуры тестирования и документирования информации.

В мировой практике обозначились два вида сертификации: мягкая и жесткая. Жесткая сертификация – это сертификация программного обеспечения, применяемого в особо ответственных приложениях (в авиационных системах, атомных электростанциях и др.). «Жесткая сертификация в обязательном порядке должна соответствовать стандартам и нормативно-технической документации той или проблемно-ориентированной области. «Мягкая» сертификация позволяет повысить экономическую конкурентоспособность программных продуктов и обеспечить необходимый уровень «интерфейсной» совместимости.

Сертификация подразделяется на несколько уровней: государственный уровень (для особо ответственных систем), ведомственный (например, для авиации) и внутрифирменный.

При сертификации программного обеспечения используется как стандарты, так и нормативно-технические документы на данный продукт, технические условия, функциональные характеристики, включающие с одной стороны существующие стандарты, а с другой – «оригинальную часть», которую рекламирует разработчик и которая должна быть проверена и удостоена сертификации.

11. Выбор и оценка программных продуктов

Повышение значимости программных продуктов и степени зависимости отдельных потребителей и всей экономики в целом от программных средств вызывает необходимость и особую актуальность проблемы выбора и оценки пригодности и качества программного продукта.

Ниже предлагается подход, который позволяет формализовать процесс выбора и оценки программного продукта, выбрать его оптимальные характеристики на основе анализа множества программных продуктов, предлагаемых на рынке, обеспечивающего выполнение заданного набора функций.

Процедура выбора и оценки программного продукта подразделяется на следующие этапы с выполнением ряда формально-логических процедур:

- определение текущих и будущих требований к обработке информации;
- обзор всех доступных программных продуктов для выполнения требований к обработке информации;
- составление перечня наиболее подходящих программных продуктов;
- оценка основных характеристик программного продукта и предварительный выбор;
- оценка качественных характеристик программного продукта;
- выбор программного продукта.

Основными характеристиками программного продукта являются; алгоритмическая сложность; состав и глубина проработки реализованных функций обработки; полнота и системность функций обработки; объем файлов программ; требования к операционной системе и техническим средствам со стороны программного продукта; требуемый объем оперативной и дисковой памяти; тип процессора и версия операционной системы.

При выборе и оценке программного продукта немаловажным фактором выступают его качественные характеристики: мобильность; надежность; эффективность; учет человеческого фактора; модифицируемость; коммуникативность.

На выбор и оценку программного продукта также оказывают влияние: страна и фирма-производитель, цена, а также затраты связанные с внедрением, сопровождением и эксплуатацией продукта.

Для формализации задачи выбора и оценки программного продукта вводятся следующие обозначения:

- M – множество программных продуктов, реализующих требования пользователей к обработке информации; i – индекс программного продукта; $i \in M = 1, M$;

- N - множество функций обработки информации, которые необходимо реализовать пользователю; j – индекс функций обработки информации, которые необходимо реализовать пользователю; $j \in N = 1, N$;

- C_{ij} – стоимость i – го программного продукта для реализации j – х функций обработки информации;

- T_{ij} – время реализации с помощью i – го программного продукта j – й функции обработки информации;

- I_{ij} – объем оперативной памяти, занимаемый i - м программным продуктом при реализации j – х функций обработки информации;

- G_{ij} - объем памяти на жестком диске, занимаемым i – программным продуктом при реализации j – х функций обработки информации;

- H_{ij} – требуемая тактовая частота, необходимая для i - го программного продукта при реализации j – х функций обработки информации;

- P_{ij} – общая производительность технических средств, необходимая для i - го программного продукта при реализации j – х функций обработки информации;

- Z_{ij} – булева переменная, равная 1, если с помощью i – го программного продукта реализуется j – е функций обработки информации и равная 0 – в противном случае.

Тогда задача выбора программного продукта заключается в следующем: найти вектор $W = \{w_{ij}\}$, минимизирующий функцию суммарных затрат пользователей.

$$\sum_{i=1}^M \sum_{j=1}^N C_{ij} * w_{ij} \rightarrow \min,$$

при:

$$\sum_{i=1}^M \sum_{j=1}^N I_{ij} * w_{ij} \leq I_a;$$

$$\sum_{i=1}^M w_{ij} = 1; \quad ; j = 1, N;$$

$$w_{ij} = 1, 0, \quad i = 1, M; j = 1, N.$$

здесь I_a – оперативный объем памяти ЭВМ.

Если необходимо учесть объем памяти на жестком диске (G), тактовую частоту (H) и производительность ЭВМ (P) при условии, что стоимость затрат не превышает величины C , то формальная постановка задачи имеет следующий вид

$$\sum_{i=1}^M \sum_{j=1}^N (G_{ij}, C_{ij}, P_{ij}) w_{ij} \rightarrow \min,$$

при

$$\sum_{i=1}^M \sum_{j=1}^N C_{ij} w_{ij} \leq C;$$

$$\sum_{i=1}^M w_{ij} = 1; \quad j = 1, N;$$

$$w_{ij} = 1, 0, \quad i = 1, M; j = 1, N.$$

Очень часто пользователь готов заплатить ту сумму, которая требуется, лишь бы минимизировать время реакции программного продукта при ограничении параметров (I), (G), (H), (P). Тогда формальная постановка задачи будет иметь вид:

$$\sum_{i=1}^M \sum_{j=1}^N G_{ij} w_{ij} \leq G_a;$$

$$\sum_{i=1}^M \sum_{j=1}^N H_{ij} w_{ij} \leq H_a;$$

$$\sum_{i=1}^M \sum_{j=1}^N P_{ij} w_{ij} \leq P_a;$$

$$\sum_{i=1}^M \sum_{j=1}^N w_{ij} = 1 \quad j = 1, N;$$

$$w_{ij} = 1, 0, \quad i = 1, M; j = 1, N.$$

здесь: G_a – допустимый объем памяти на жестком диске; H_a – допустимая тактовая частота; P_a – максимальная производительность технических средств (ЭВМ).

Если минимизации подлежит объем оперативной памяти ЭВМ, занимаемый программным продуктом при условии, что стоимость затрат не превышает величины C , то формальная постановка задачи имеет следующий вид

$$\sum_{i=1}^M \sum_{j=1}^N I_{ij} * w_{ij} \rightarrow \min,$$

при

$$\sum_{i=1}^M \sum_{j=1}^N C_{ij} w_{ij} \leq C;$$

$$\max \sum_{i=1}^M \sum_{j=1}^N T_{ij} * w_{ij} \rightarrow \min,$$

при

$$\sum_{i=1}^M \sum_{j=1}^N I_{ij} * w_{ij} \leq I_a;$$

$$\sum_{i=1}^M w_{ij} = 1; \quad j = 1, N;$$

$$w_{ij} = 1, 0, \quad i = 1, M; \quad j = 1, N.$$

На основании решения перечисленных задач осуществляется предварительный выбор и оценка программного продукта, учитывающие его основные технические характеристики. Окончательный выбор программных продуктов осуществляется с учетом его качественных характеристик (мобильности, надежности, эффективности, учета человеческого фактора, коммуникативности), а также с учетом страны и фирмы производителя программного продукта.

Одним из методов оценки качественных характеристик является метод экспертных оценок, по которому показатели качества программного продукта определяются по формуле:

$$A_i = \sum_{k=1}^K a_{ik},$$

где A_i – интегральная оценка качества i – го программного средства; a_{ik} – оценка качества i – го программного средства по k – му показателю качества; $k = 1, K$.

Технологически методику оценки качества программных продуктов целесообразно реализовать на базе методики экспертных оценок.

Следует отметить, что любой программный продукт может оцениваться только применительно к конкретной задаче и никоим образом не сам по себе. Программные продукты, предназначенные для внедрения в качестве базиса информационных систем, обладают одним общим характерным свойством: они сложны для оперативного ознакомления. Эта проблема обусловлена следующими факторами:

- сложность не только внутренних механизмов работы, но и наблюдаемой функциональной структуры.
- большой набор специфических инструментов для различных областей менеджмента. Например, многие производственные и технологические тонкости неизвестны финансовому директору, и наоборот, главный инженер некомпетентен в принципах анализа финансовых отклонений. Специалист, принимающий решения по выбору программного комплекса, как правило, является IT – менеджером и имеет лишь общее и неполное представление об использующихся управленческих методиках.
- наличие специальной терминологии, большого количества стандартов и псевдостандартов информационного менеджмента.

- доступность материалов исключительно рекламного характера, фактическое отсутствие описания реального опыта использования программного комплекса и истинной статистики внедрения.

Многие зарубежные программные продукты имеют солидный и позитивный опыт применения на Западе. Однако не стоит забывать, что сами по себе подходы к управлению в нашей стране и на Западе существенно различаются. Например, в большинстве развитых экономических стран существуют и широко применяются на практике отраслевые стандарты менеджмента. Тем самым, западные тиражируемые ПП, как правило, подразумевают наличие общего регламента управления деятельностью предприятия, при этом, позволяя (благодаря широким возможностям по настройке) учитывать все индивидуальные особенности. То же самое можно отнести и к понятию «отраслевое решение». Не секрет, что практически не существует отраслевых управленческих стандартов (имеются в виду именно управленческие, а не технологические стандарты), и два предприятия, относящейся к одной отрасли, могут принципиально различаться с точки зрения действующего управленческого регламента.

Несомненно, комплексные зарубежные решения применимы и у нас. Более того, при правильном подходе, их использование будет не менее продуктивным, чем на западе. Однако, для того, чтобы их внедрение было успешным, всегда необходимо осуществлять реорганизацию бизнес процессов, разрабатывать и утверждать регламент всех процедур и алгоритмов. Известно, что такой подход не является дешевым, однако ошибочно в целях экономии избегать его и вкладывать миллионы долларов в неэффективную информационную систему, пытаясь настроить, например, настроить подсистему производственного планирования в тех случаях, когда сама процедура планирования на предприятии не регламентировано и де-факто не существует.

Необходимость оценки качества программных продуктов следует рассматривать со следующих позиций:

1. Эффективности функционирования индустрии программного обеспечения в широком смысле. В этом случае показатели качества характеризуют общий уровень различных классов программных продуктов, и дает информацию для подготовки рекомендаций об их дальнейшем развитии.

2. Повышение информированности пользователей о качестве программных продуктов с целью более активного их влияния на повышение эффективности и качества программных продуктов.

3. Повышение обоснованности принимаемых решений при приобретении программных продуктов.

12. Категория специалистов, связанных с созданием и эксплуатацией программ

Основная категория специалистов, занятых разработкой программ – это программисты (programmer). Программисты неоднородны по уровню

квалификации, а также по характеру своей деятельности. Наиболее часто программисты делятся на системных программистов, прикладных и аналитиков.

В создании и эксплуатации программ также участвуют постановщик задачи, администратор базы данных, конечный пользователь (рис 5.).

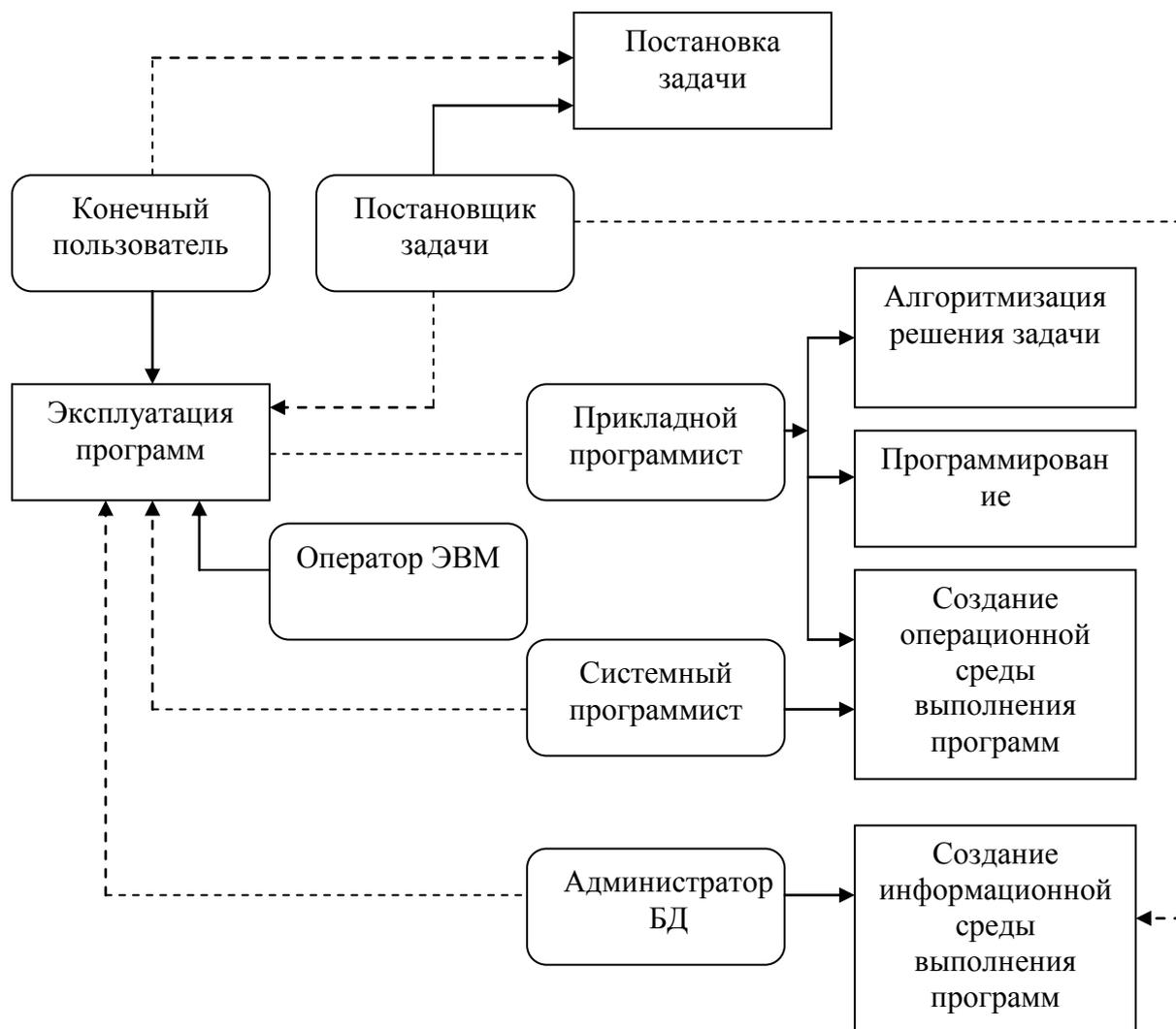


Рис 5. Схема взаимодействия специалистов связанных с созданием и эксплуатацией программ

Системный программист (system\ software programmer) занимается разработкой, эксплуатацией и сопровождением системного программного обеспечения, поддерживающего работоспособность компьютера и создающего среду для выполнения программ, обеспечивающих реализацию функциональных задач.

Прикладной программист (application programmer) осуществляет разработку и отладку программ для решения функциональных задач.

Программист-аналитик (programmer- analyst). Это новая квалификация программистов, которая появляется в условиях создания

больших по масштабам и функциям обработки программ, который анализирует и проектирует комплекс взаимосвязанных программ для реализации функций предметной области (совокупности функциональных задач).

Постановщик задачи участвует в процессе создания программ на начальной стадии работ.

Администратор базы данных выполняет работы связанных с организационной поддержкой базы данных. Такая необходимость возникает, если база данных является интегрированной, обеспечивающей работу с данными многих приложений.

Конечный пользователь (end user) является основным потребителем программ и относится, как правило, к категории пользователей – непрограммистов. Конечный пользователь не является специалистом в области программирования, т. е. не владеет методами и технологией проектирования и создания программ, но имеет элементарные знания и навыки работы с вычислительной техникой. Такая квалификационная характеристика пользователя программного обеспечения в значительной степени влияет на спецификацию требований к создаваемым программам, интерфейсам, формам машинных документов, технологии решения задачи на ЭВМ.

Возможна эксплуатация программ квалифицированными программистами или специально обученными техническими работниками – операторами ЭВМ. Взаимодействие специалистов различного вида, участвующих в создании и эксплуатации программ, показано на рис. В ряде случаев один специалист совмещает несколько видов деятельности.

Администратор базы данных и системный программист осуществляют подготовку информационных и программно-технических условий для работы программ. Пунктирные линии показывают участие специалиста в качестве консультанта.

13. Правовые методы защиты программных продуктов

Спецификой программных продуктов является то, что их разработка, использование и эксплуатация должна выполняться на правовой основе.

Правовые методы защиты программных продуктов включают:

- патентную защиту;
- лицензионные соглашения и контакты;
- закон о производственных секторах;
- закон об авторском праве.

Различают две категории прав:

- экономические права, дающие их обладателям право на получение экономических выгод от продажи или использования программных продуктов;

- моральные права, обеспечивающие защиту прав личности автора в его произведениях.

Во многих цивилизованных странах несанкционированное копирование программ в целях продажи или бесплатного распространения рассматривается как государственное преступление, карается штрафом или тюремным заключением. Но, к сожалению, само авторское право не обеспечивает защиту новой идеи, концепции, методологии и технологии разработки программ, поэтому требуются дополнительные меры их защиты.

Патентная защита устанавливает приоритет в разработке и использовании нового подхода или метода при разработке программ, удостоверяет их оригинальность.

Статус производственного сектора для программы ограничивает круг лиц, знакомых или допущенных к ее эксплуатации, а также определяет меру их ответственности за разглашение секретов. Например, используется парольный доступ к программному продукту или базе данных, вплоть до паролей на отдельные режимы (чтение, запись, корректировку и т. д.). Программы, как объект большой стоимости, необходимо охранять от кражи и преднамеренных нарушений.

Лицензионные соглашения распространяются на все аспекты правовой охраны программных продуктов, включая авторское право, патентную защиту, производственные секреты. Наиболее часто используются лицензионные соглашения на передачу авторских прав.

Лицензия – договор на передачу одним лицом (лицензиаром) другому лицу (лицензиату) права на использование имени, продукции, технологии или услуги. Лицензиар увеличивает свои доходы сбором лицензионных соглашений, расширяет область распространения программного продукта; лицензиат извлекает доходы за счет их применения.

В лицензионном соглашении оговариваются все условия эксплуатации программ, в том числе создание копий. На каждой копии программы должны быть те же самые отметки, что и на оригинале:

- знак авторского права (обычно ©) и название разработчика, года выпуска программы, прочих ее атрибутов;
- знак патентной защиты или производственного сектора;
- торговые марки, соответствующие использованным в программе другим программным изделиям (обычно ™ и название фирмы разработчика программного продукта);
- символ зарегистрированного права на распространение программного продукта (обычно ®).

Существует несколько видов лицензии на программные продукты.

Исключительная лицензия – продажа всех имущественных прав на программный продукт. Покупателю лицензии предоставляются исключительное право на их использование, а автор или владелец патента отказывается от самостоятельного их применения или предоставления другим лицам.

Это самый дорогой вид лицензии, к нему прибегают для монопольного владения, либо с целью извлечения дополнительной прибыли, либо с целью прекращения существования рынка программных средств программного продукта.

Простая лицензия - лицензиар предоставляет право лицензиату использовать программный продукт, оставляя за собой право применять их и предоставлять на аналогичных условиях неограниченному числу лиц (лицензиат при этом не может сам выдавать сублицензии, может лишь продавать копии приобретенного программного продукта).

Такой вид лицензии приобретают дилер (торговец), либо фирмы-производители, использующие купленные лицензии как сопутствующий товар к основному виду деятельности. Например, многие производители и фирмы, торгующие вычислительной техникой, осуществляют продажу вычислительной техники с установленным лицензионным программным обеспечением (операционная система, текстовый редактор, электронная таблица, графические пакеты и т. д.).

Этикеточная лицензия – лицензия на одну копию программного продукта. Данный тип лицензии применяется при розничной продаже. Каждый официальный покупатель заключает лицензионное соглашение с продавцом на их использование, но при этом сохраняется авторское право разработчика.

Экономические отношения между лицензиаром и лицензиатом могут строиться различным образом (рис 6.). За правом пользования программным продуктом выплачивается единовременное вознаграждение (паушальный платеж), которое и является фактической ценой лицензии. Возможны и периодические отчисления лицензиару за право пользования в виде роялти – фиксированная ставка в определенные интервалы времени в течение действия лицензионного соглашения, как правило, процент от стоимости программных продуктов.

Закон об охране программных продуктов автором признает физическое лицо, в результате творческой деятельности которого они созданы. Автору, независимо от его имущественных прав принадлежат личные авторские права: авторство, имя, неприкосновенность (целостность) программ.

Пользователь	Права	Деньги	Программный продукт
Автор программы Обладатель исключительной лицензии Обладатель простой лицензии Покупатель этикеточной лицензии	↓	↑	↓

Рис 6. Движение прав, денег на программные продукты при лицензионном соглашении.

Авторское право действует с момента создания программного продукта в течение всей жизни автора и, после его смерти. В течение определенного периода времени автор может:

- выпускать в свет;
- воспроизводить в любой форме, любыми способами;
- распространять;
- модифицировать
- осуществлять любое другое использование программного продукта.

Авторское право не связано с правом собственности на материнский носитель.

Имущественные права на программный продукт могут быть переданы частично или полностью другим физическим или юридическим лицам по договору. Если программный продукт создан в порядке служебных обязанностей, имущественные права принадлежат работодателю.

Программные продукты могут использоваться третьими лицами – пользователями на основании договора с правообладателем.

Лицо, правомерно владеющее экземпляром программ вправе, без получения дополнительного решения правообладателя, осуществлять любые действия, связанные с функционированием программного продукта в соответствии с ее назначением, в том числе:

- устанавливать один экземпляр, если не предусмотрено иное соглашение с правообладателем программного продукта на компьютер;
- исправлять явные ошибки;
- адаптировать программный продукт;
- изготавливать страховые копии.

Заключение

В современных рыночных условиях ни одно крупное и даже маленькое предприятие не может обойтись без услуг информации и информационно-коммуникационных технологий. Информация, в наше время, занимает огромное место в экономике всех стран. С каждым днем разрабатываются все новые и новые компьютерные программы, которые облегчают получение, обработку и работу с новой информацией. Поэтому вопросы, связанные с разработкой, сертификацией, выводом на рынок программных продуктов, их оценка и выбор с учетом конкретных требований и условий применения приобретают все большее значение.

Кейс-стади. « Проектирование алгоритмов и программ»

С использованием подхода «Информационное моделирование предметной области и связанных с ней приложений» провести проектирование алгоритмов и программ информационной системы «Супермаркет».

1. Постановка задачи.

Целью построения информационной системы «Супермаркет» является сбор, хранение, обработка и распространение между потенциальными пользователями информации о функционировании, как-то: предоставление руководителю отчета о деятельности супермаркета, о процессе приема заявок, регистрации и анализа документации, донесение до клиентов и работников фирмы информации о результатах деятельности супермаркета.

В соответствии с поставленной целью построение информационной системы ориентировано на решение следующих задач:

- отразить процесс организации поступления продукции в супермаркете;
- детально рассмотреть процедуру поставки в супермаркет продукции, как одну из основных стадий в его успешной работе;
- представить количественные и качественные характеристики функционирования супермаркета для выявления эффективности его деятельности путем составления запросов;
- отразить отчетность по функционированию супермаркета через представление запросов и отчетов.

Решение задачи

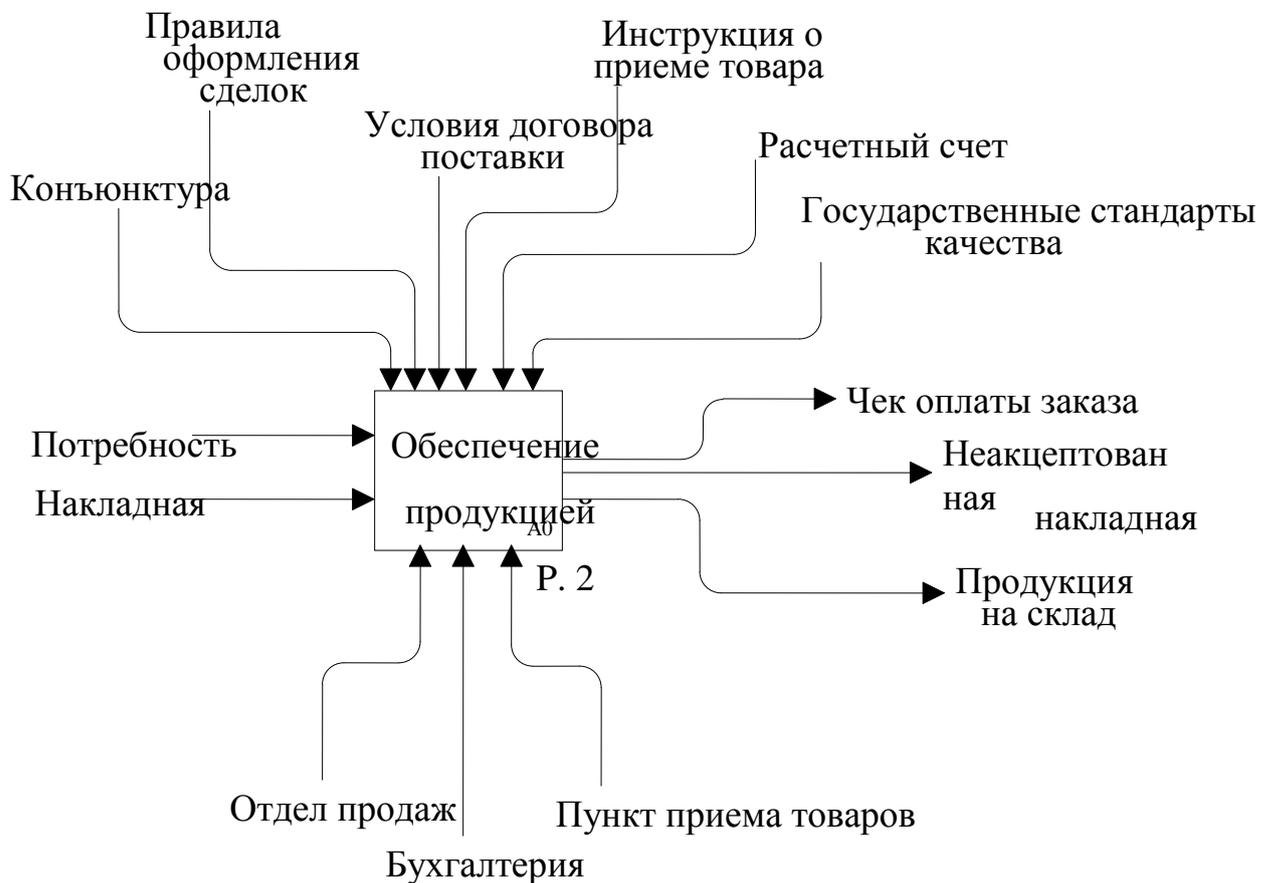
В основе информационной системы «Супермаркет» лежит динамически обновляемая модель предметной области на основе базы данных (БД).

Для реализации поставленных задач используется Desing Idef и система управления базой данных СУБД Access. Проектирование информационной системы «Супермаркет» осуществляется средствами системно-структурного анализа и теорией реляционной БД.

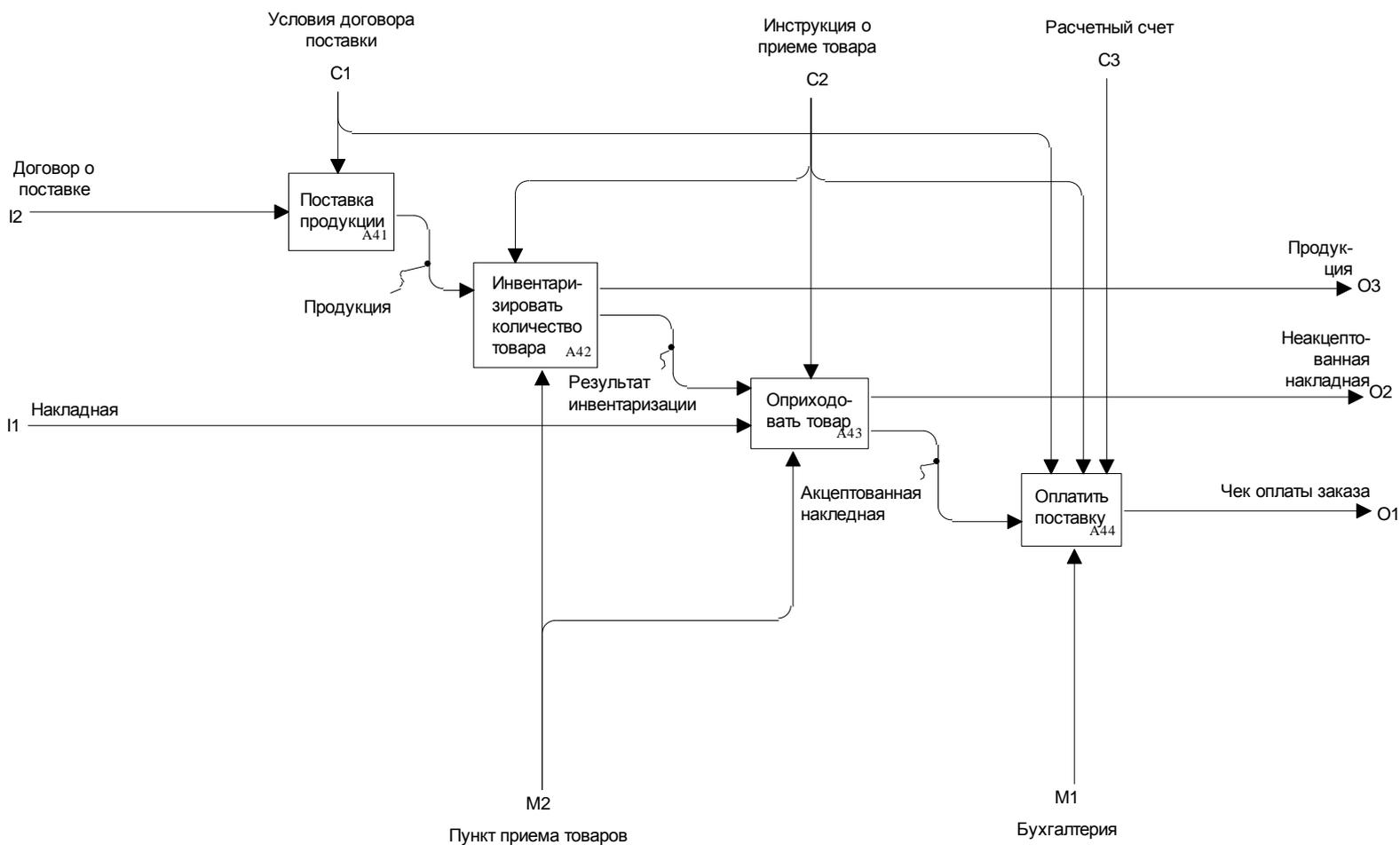
Построение функциональной модели предметной области.
 Проектирование информационной системы осуществляется путем построения функциональной модели при помощи Desing Idef.

а) Описание деятельности супермаркета в общем виде (создание контекстной диаграммы).

Положив в основу функционального блока модели этап поставки продукции в супермаркет, указываем входящие и выходящие (новые или преобразованные старые) объекты информационной системы; управленческие аспекты, ограничивающие функционирование информационной системы; механизмы, с помощью которых выполняется процесс.

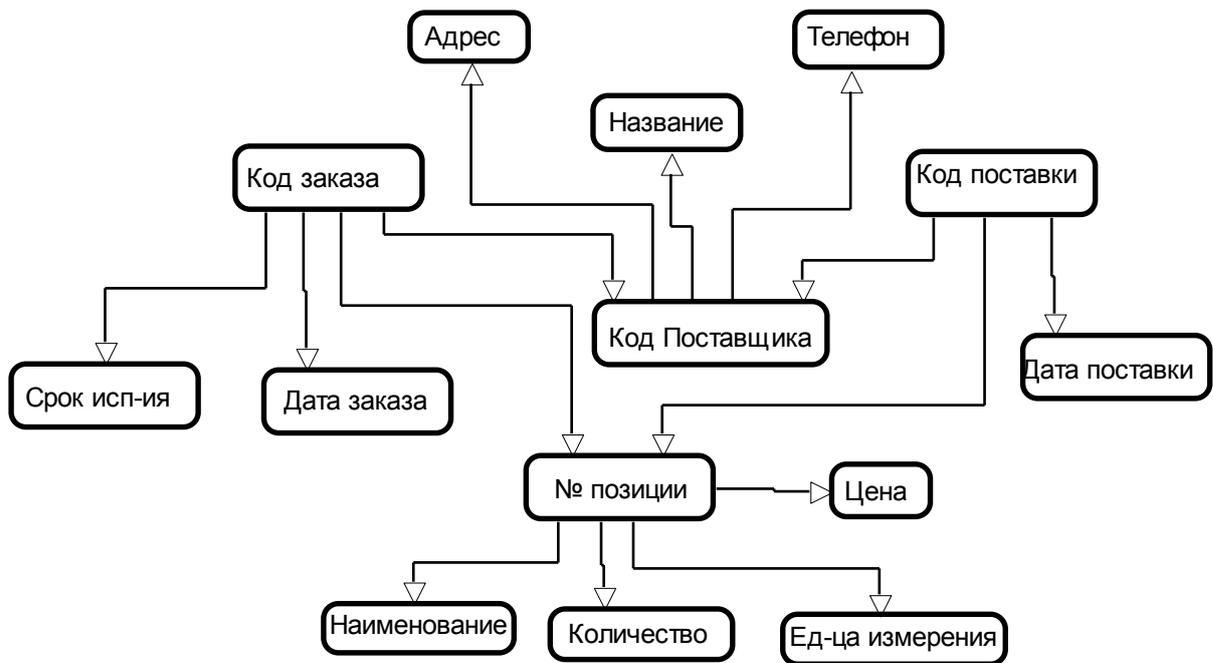


б) Путем дальнейшей декомпозиции детально рассмотрим процесс поставки и расчета за продукцию, от которого полностью зависит эффективное и регулярное пополнение запасов и ассортимента, то есть результат функционирования супермаркета.

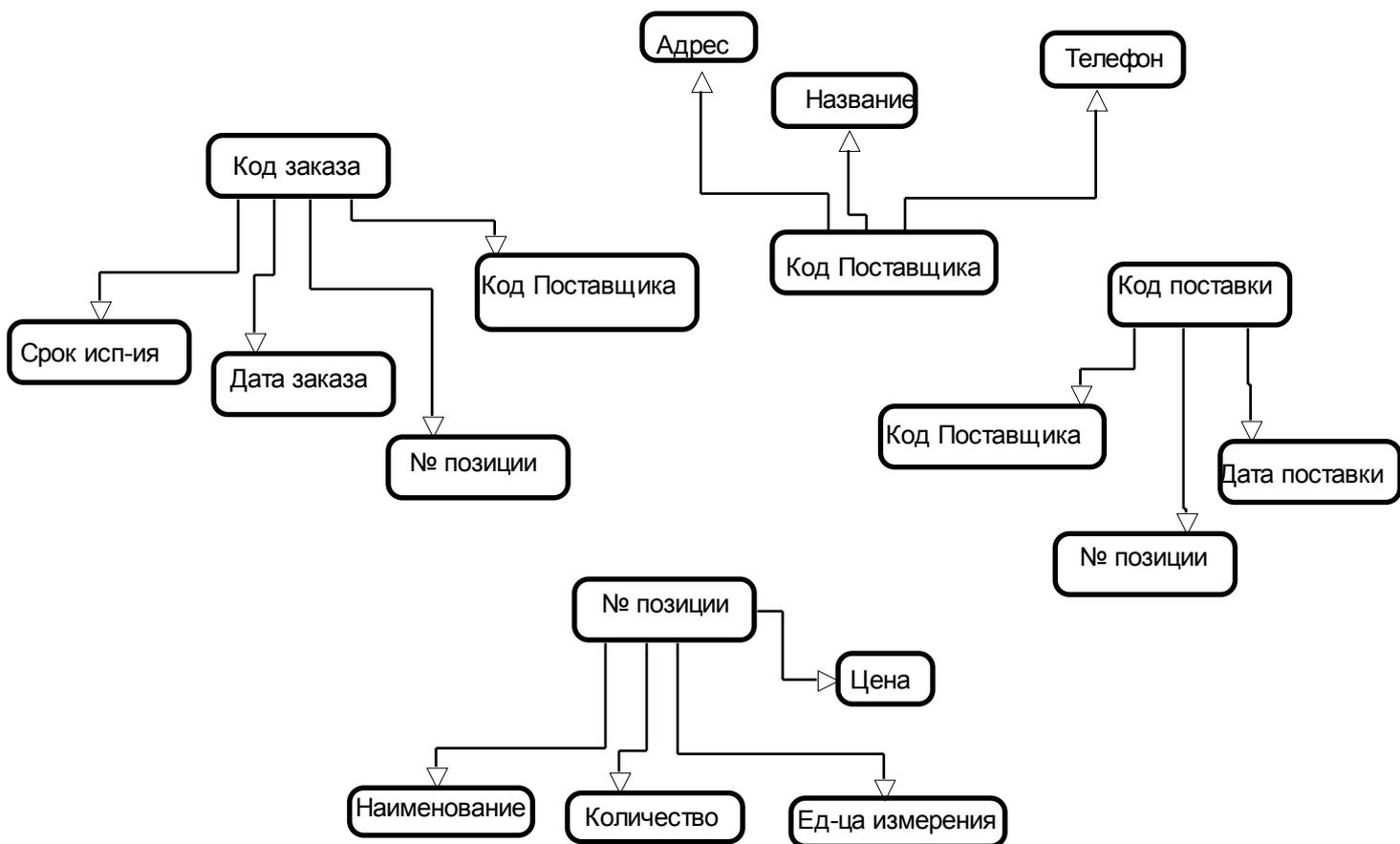


Таким образом, информационная система «Супермаркет» демонстрирует, что все входящие объекты, связанные с обеспечением продукцией и необходимые для работы супермаркета, в результате функционирования информационной системы пройдут в соответствии со стандартами и правилами, условиями договора поставки и инструкциями регистрацию и анализ и преобразуются в конечные результаты, необходимые для эффективной организации деятельности супермаркета.

Датологическое проектирование. Исходя из построенной функциональной модели, начинаем процесс создания функциональных отношений. Для этого из ИС «СМ» выделяются главные атрибуты, описывающие данную предметную область, то есть деятельность СМ по оформлению заявок, поиску поставщиков, заключению договора и, наконец, поставке продукции. Между атрибутами устанавливаются функциональные зависимости, в результате чего получаются универсальные отношения, содержащие всю информацию о сформированной БД.

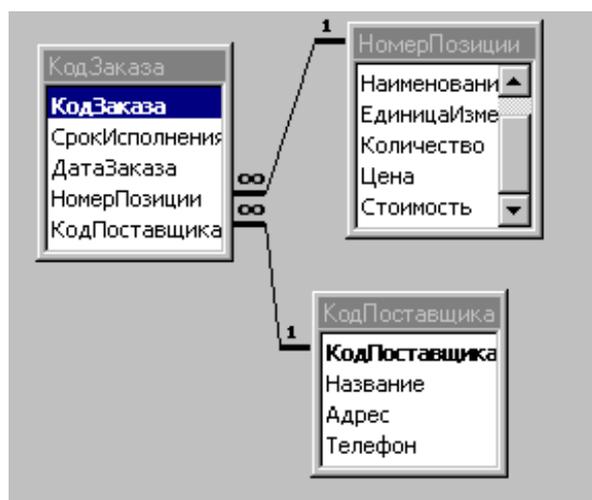


Далее осуществляется декомпозиция универсальных отношений до тех пор, пока каждое из отношений не будет находиться в изолированной форме.



Создание информационной системы средствами СУБД Access. На основе полученной декомпозиции универсальных отношений формируется БД для построения таблиц. По модели «сущность-связь» устанавливаем связи и заполняем уже созданные таблицы. В результате получаем всю информацию БД «Супермаркет», на основе которой строим запросы, систематизирующие необходимые сведения.

Для начала необходимо построить схему данных, отражающую имеющиеся связи.



Затем можно приступать к построению запросов. Наиболее простым из них является запрос " Кто поставляет продукт", который выдает названия поставщиков основных продуктов супермаркета и имеет следующий вид:

Поле:	Наименование	Название
Имя таблицы:	НомерПозиции	КодПоставщика
Сортировка:		
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:		
или:		

Следующий запрос приводит данные о том, у какого поставщика, по какому адресу и телефону можно достать конкретный продукт, если в нем появится необходимость. Запрос имеет следующий вид.

Поле:	НомерПозиции	Наименование	КодПоставщика	Название	Адрес	Телефон
Имя таблицы:	КодЗаказа	НомерПозиции	КодЗаказа	КодПоставщика	КодПоставщика	КодПостави
Сортировка:						
Вывод на экран:	<input checked="" type="checkbox"/>					
Условие отбора:		"макароны"				
или:						

Можно также строить и более сложные запросы. Таким, например, является запрос, который позволяет проследить динамику заказов по числам, то есть показывает количество заказов супермаркета по дням.

Поле:	ДатаЗаказа	Наименование
Имя таблицы:	КодЗаказа	НомерПозиции
Групповая операция:	Группировка	Count
Сортировка:		
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:		
или:		

Можно также проследить объем поставок по дням, то есть создать запрос, который подсчитывает количество поставляемой продукции по дням.

Поле:	СрокИсполнения	Наименование	ЕдиницаИзмерени	Количество
Имя таблицы:	КодЗаказа	НомерПозиции	НомерПозиции	НомерПозиции
Групповая операция:	Группировка	Группировка	Группировка	Sum
Сортировка:				
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:				
или:				

Таким образом, Microsoft Access позволяет строить любые запросы и получать всю необходимую информацию об имеющейся совокупности.

Вопросы и задания

1. Какими важнейшими признаками классифицируются методы проектирования алгоритмов и программ?
2. В чем суть неавтоматизированного проектирования алгоритмов и программ?
3. В чем суть автоматизированного проектирования алгоритмов и программ?
4. Какие подходы проектирование алгоритмов и программ наиболее распространены?

5. В чем заключается суть структурного проектирования, и какими методами оно представлено?
6. В чем заключается суть метода проектирования «Информационное моделирование предметной области и связанных с ней приложений» и при каких условиях развития программных средств он появился?
7. В чем заключается суть объектно-ориентированного подхода?
8. Что является целью построения информационной системы «Супермаркет»?
9. Объясните, почему в качестве метода проектирования выбран метод «Информационное моделирование предметной области и связанных с ней приложений»?
10. Из каких работ состоит проектирование информационной системы «Супермаркет»?

Использованная литература

1. Гулямов С.С., Мусалиев А.А., Бегалов, Б.А. Информационные системы, технологии и организация. Методический материал по предмету «Проектирование экономических информационных систем». - Т.: ТГЭУ, 2005.
2. Мусалиев А.А. Информационный бизнес: Тексты лекций. – Т.: ТГЭУ, 2005.
3. Миллий иктисодда ахборот тизимлари ва технологиялари: Олий уқув юртлири талабалари учун уқув кулланма // Муаллифлар: Р.Х.Алимов, Б.Ю.Ходиев, К.А.Алимов ва бошк.; С.С.Гуломовнинг умумий тахрири остида. – Т.: «Шарк», 2004.
4. Ходиев Б.Ю., Мусалиев А.А., Бегалов Б.А. Введение в информационные системы и технологии : Учебное пособие. – Т.: ТГЭУ, 2002.
5. Оценка и аттестация зрелости процессов создания и сопровождения программных средств и информационных систем. Пер. с англ. А.С. Аипова и др.- М.: Книга и бизнес, 2001.
6. Охунов Д.М. Исследование и разработка маркетинговых автоматизированных информационных систем предприятий. Диссертация на соискание степени кандидата экономических наук. Ташкент, ТГЭУ, 2005, 138 с.
7. Вендров А.М. Проектирование программного обеспечения экономических информационных систем: Учебник. – М.: Финансы и статистика, 2000.