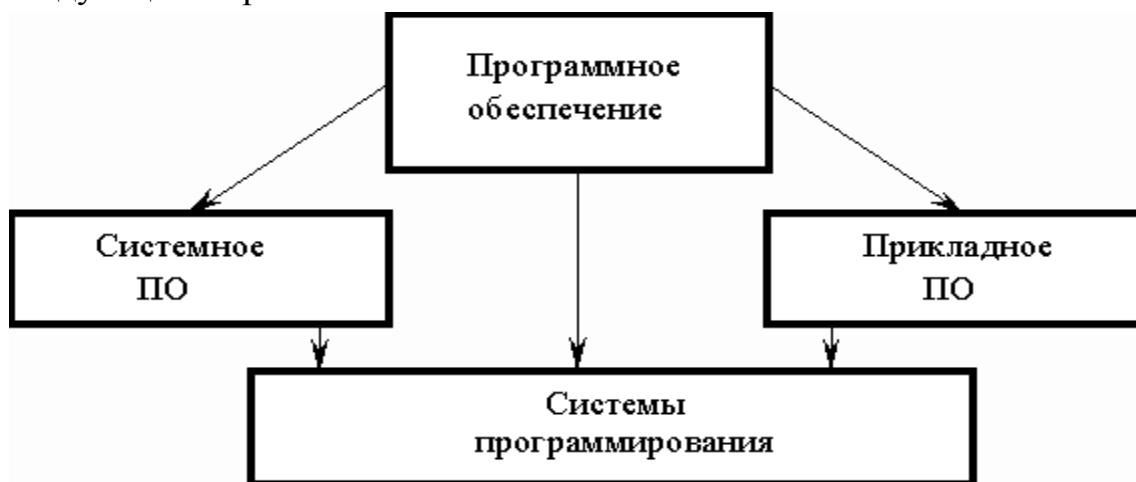


Эволюция программного обеспечения

*Левашова Мария
студентка группы В 42
научный руководитель:
старший преподаватель
кафедры математики,
информатики и естествознания
Иманмухаметова Б.М.*

Программное обеспечение (ПО) - это определенная совокупность всех программ установленных на ЭВМ, которая обеспечивает использование ЭВМ в интересах каждого ее пользователя.

Различают системное и прикладное программное обеспечение. В виде схемы классификацию программного обеспечения можно представить следующим образом:



Системное ПО – это некая совокупность программ, обеспечивающих работу компьютера. Системное программное обеспечение подразделяется на сервисное и базовое. Системные программы используются для управления работой вычислительной системы, также выполняют различные вспомогательные функции (копирования, форматирования, тестирования и т. д).

Базовое программное обеспечение включает:

- Различные операционные системы;
- Программные оболочки;
- сетевые операционные системы.
- Сервисное программное обеспечение включает в себя следующие утилиты:
- Утилиты диагностики;

- Антивирусные программы;
- Утилиты обслуживания носителей;
- Утилиты архивирования;
- Утилиты обслуживания сети.

Прикладное программное обеспечение – это совокупность программ, решающая задачи определённого класса определенной предметной области.

Прикладное программное обеспечение работает только при наличии системного ПО.

Прикладные программы называются приложениями. Они включают следующие компоненты:

- 1 текстовые процессоры;
- 2 экспертные системы;
- 2 табличные процессоры;
- 4 базы данных;
- 5 системы иллюстративной и деловой графики (графические процессоры);
- игры;
- 6 обучающие программы;
- 7 программы математических расчетов, моделирования и анализа;
- 8 интегрированные пакеты;
- 9 коммуникационные программы.

Отдельную группу составляют системы программирования (инструментальные системы), которые выступают частью системного ПО, но носят практический характер. Системы программирования – это некая совокупность программ, предназначенных для разработки, отладки и внедрения новых программ. Системы программирования обычно включают в себя:

- 1) среду разработки программ;
- 2) библиотеки справочных программ (функций, процедур);
- 3) трансляторы;
- 4) отладчики;
- 5) редакторы связей и др.

Процесс проектирования информационных систем описывает процесс создания и сопровождения программных систем в виде жизненного цикла инструментальной системы, представляя его как некую последовательность этапов и выполняемых процессов. Для каждой стадии определяются состав и линейный алгоритм выполняемых работ, получаемые конечные результаты, методы и средства, необходимые для выполнения работ, роли и ответственность участников и т.д. Такое формальное описание жизненного

цикла ИС позволяет планировать и организовать процесс разработки и обеспечить управление данным процессом.

Жизненный цикл инструментальной системы можно представить как ряд событий, происходящих с системой в процессе ее создания и использования.

Модель жизненного цикла отражает множественные состояния системы, начиная с момента возникновения данной ИС и заканчивая моментом полного выхода ИС из употребления. Модель ЖЦ - структура, содержащая процессы, действия и задачи, которые осуществляются в ходе процесса разработки, функционирования и сопровождения программного средства в течение всей жизни ИС, от определения требований до завершения использования.

В настоящее время используются следующие модели ЖЦ:

Каскадная модель ([рис. 1](#)), она предусматривает последовательное выполнение всех этапов проекта в строго определенном порядке.

Переход на следующий второй этап означает полное окончание работ на первом этапе.

Поэтапная модель с периодическим контролем ([рис.2](#)). Разработка ИС ведется с циклами установки связи между двумя этапами. Межэтапные изменения позволяют систематизировать реально существующее взаимовлияние результатов разработки на различных этапах; время жизни каждого этапа распространяется на весь период разработки.

Спиральная модель([рис.3](#)) предусматривает наличие спиралей. На каждом витке некоторой спирали выполняется создание следующей версии продукта, выдвигаются требования проекта, определяются качественные характеристики и планируются работы следующего витка. Особое внимание уделяется первым этапам разработки - анализу и проектированию, где реализуемость технических решений проверяется и обосновывается посредством создания моделей (макетирования).



Рис.1 Каскадная модель ЖЦ ИС



Рис.2 Поэтапная модель с промежуточным контролем

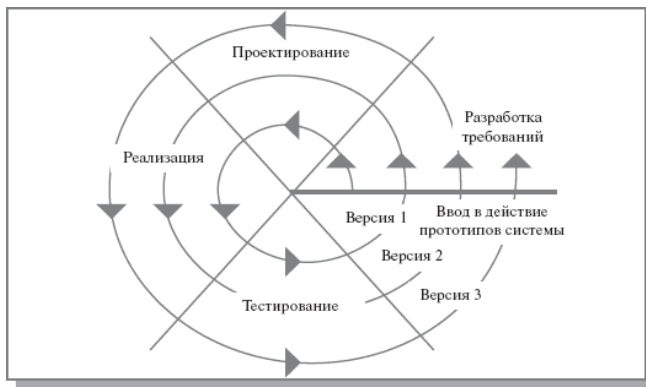


Рис.3 Спиральная модель ЖЦ ИС

Наибольшее распространение получили следующие основные модели жизненного цикла:

- каскадная модель (характерна для периода с 1970-1985 гг.);
- спиральная модель (характерна для периода после 1986года.).

В ранних проектах достаточно простых ИС приложение представляло собой единую, функционально и информационно независимую систему. Для создания такого типа приложений эффективным оказался каскадный метод. Каждый этап завершался после полного выполнения и документального оформления всех предусмотренных работ.

Можно выделить следующие стороны применения каскадного подхода:

- на каждом этапе создается законченный набор проектной документации, отвечающий требованиям полноты и валидности;
- выполняемые последовательно этапы работ позволяют спланировать сроки завершения всех выполняемых работ и соответствующие расходы.

Каскадный подход хорошо зарекомендовал себя при построении довольно таки простых ИС, когда в самом начале разработки можно достаточно точно и полно представить все требования к системе. Основным

недостатком этого подхода является то, что действительный процесс создания системы никогда полностью не укладывается в такую ограниченную схему, постоянно возникает потребность в возврате к начальным этапам и уточнении или пересмотре ранее принятых решений. В результате реальный процесс создания ИС оказывается соответствующим поэтапной модели с периодическим контролем.

Согласование результатов пользователями производится в тех точках, планируемых после завершения каждого этапа работы, а общие требования к ИС записаны в виде технического задания на все время ее написания. Однако и эта схема не позволяет своевременно принимать возникающие изменения и уточнения требований к системе.

Таким образом, пользователи чаще всего получают систему, не удовлетворяющую их требованиям.

Спиральная модель ЖЦ была предложена как раз ради преодоления перечисленных проблем. На этапах анализа и проектирования возможность реализации технических решений и степень удовлетворения потребностей покупателя проверяется путем создания прототипов. Каждый виток спирали соответствует созданию работающего фрагмента или оптимальной версии системы. Это помогает уточнить требования, цели и характеристики проекта, определить качество разработки, спланировать работы следующего витка системы.

Таким образом, углубляются и последовательно конкретизируются подробные детали проекта и в результате выбирается обоснованный вариант, который удовлетворяет всем требованиям заказчика и доводится до реализации.

Итеративная разработка в некоторой степени отражает объективно существующий спиральный цикл создания простых и сложных систем. Она позволяет переходить на следующий этап, не дожидаясь полного завершения работы на предыдущем и решить главную задачу - быстрее презентовать пользователям системы работающий продукт, тем самым активизируя процесс уточнения и дополнения требований.

Главная проблема спирального цикла - определение момента перехода на следующий этап. Для ее решения вводятся ограничения во времени на каждый из этапов жизненного цикла, и переход осуществляется в соответствии с алгоритмом, даже если не вся работа закончена.

Планирование производится на основе данных статистики, полученных в предыдущих проектах, и личного опыта разработчиков-программистов.

В 1953 году основатель теории информации великий американский математик К.Шеннон писал: «Наши ВМ выглядят как ученые-схоластики.

При вычислении длительной цепи арифметических действий ЦВМ значительно обгоняют человека. Когда же пытаются адаптировать ЦВМ для выполнения нематематических действий, они оказываются неуклюжими и неприспособленными для такой деятельности».

1-ый Этап: машинные ресурсы. Выдвинутые К.Шенноном функциональные ограничения и высокая стоимость первых ЭВМ в целом определяли основную задачу информационной технологии 50-х гг. - повышение эффективности форматирования данных по уже формализованным или легко преобразуемым схемам.

Основной целью выступало – сокращение общего числа машинных тактов, требовавших для своего решения конкретная программа, а также объем занимаемой ею оперативной памяти. Основные затраты на форматирование данных находились в прямой зависимости от затраченного на них времени работы машины.

2-ой Этап: программирование. В середине 60-х годов 20 века начался следующий этап развития информационных технологий, который продолжался до 80-ых годов. От технологии эффективного выполнения программ к технологии эффективного написания программ – так можно было определить общее направление изменения критериев эффективности в течение этого этапа.

Самым известным результатом этого первого критичного оценивания критериев технологии программирования стала создание в начале 70-ых годов операционной системы UNIX. Систему UNIX, нацеленную на повышение эффективности работы программистов, разработали сотрудники компании «БеллЛэбс» К.Томпсон и Д.Ритчи, которых определенно не удовлетворяли имеющиеся простейшие средства создания программ, ориентированные на пакетный режим. На рубеже 80-ых годов операционная система UNIX рассматривалась как классический образец ОС, она начала свое триумфальное шествие на мини ЭВМ серии PDP11 в середине 70-ых годов 20 века.

3 Этап: формализация данных. Персональный компьютер, чаще всего, имеет хорошо развитые средства самообучения новичка работе за машиной, мобильные средства защиты от ошибок и, что самое главное, все аппаратные и программные компоненты такой ЭВМ подчинены одной главной задаче - обеспечить «положительную реакцию» машины на любые, в том числе неадекватные, действия пользователя-новичка. Главная задача персональных вычислений- формализация профессиональных умений, выполняемая, чаще всего, самостоятельно новичком-пользователем или при минимальной технической поддержке программиста-техника.

Литература

- 1 Баженова Л.М. Медиаобразование школьника (1-4 классы). М.: Изд-во Ин-та художественного образования Российской Академии образования, 2004.
- 2 Информатика в начальном образовании: Рекомендации. А.Л.Семенов, А. Книецингер. М.: ИИТО, ИНТ, 2000.
- 3 Концепция информатизации образовательного процесса в системе Департамента образования Москвы. М. 2009
- 4 ЛогоМиры 3.0. Справочное пособие. М.: ИНТ, 2008
- 5 Пейперт С. Переворот в сознании: дети, компьютеры и плодотворные идеи. М.: Педагогика, 1989
- 6 "ПервоЛого 3.0. Справочное пособие.М.: ИНТ, 2008
- 7 Яковлева Е. И. ЛогоМозаика. Сборник проектов. М: ИНТ, 2005