

кафедра информационно-измерительных систем (ИС)



дисциплина: «Языки программирования»

Лекция 3:

«Жизненный цикл современной программной системы»

1. Разделение труда при разработке программ на ЯП (исторический взгляд)
2. Понятие жизненного цикла программной системы
3. Этапы разработки (технология) программных систем
 - 3.1. Анализ и постановка задачи
 - 3.2. Проектирование программной системы
 - 3.3. Разработка и отладка модулей программной системы
 - 3.4. Интеграция и тестирование
 - 3.5. Документирование
 - 3.6. Внедрение: установка, настройка и обучение пользователей
 - 3.7. Эксплуатация и сопровождение программной системы

1. Разделение труда при разработке программ на ЯП (исторический взгляд)

I этап: конец 1940-х – середина 1950-х гг. – машинные и машинно-ориентированные языки (языки ассемблера: 1954 – IPL и 1955 – FLOW-MATIC): компьютеры – дорогостоящие, к ним допускали лучших инженеров и ученых, зачастую, самих создателей ЭВМ, они же - **авторы программ, которые самостоятельно выполняли весь комплекс по разработке программы:**

- 1) формулировал постановку задачи;
- 2) разрабатывал алгоритм решения;
- 3) записывал на используемом на данной машине ЯП;
- 4) производил отладку и тестирование программы;
- 5) эксплуатировал разработанную программу;
- 6) при необходимости обновлял (вносил изменения) программу;
- 7) прекращал пользоваться программой (время эксплуатации программы, как правило, недолгое, исчислялось месяцами, годом использования)

1. Разделение труда при разработке программ на ЯП (исторический взгляд)

II этап: середина 1950-х (Fortran) –1970-ые гг. (Паскаль – 1970 г; С – 1972 г) – императивные языки программирования: компьютеры – более доступные, ряды программистов ширились – усложнялись решаемые задачи и размер программ (до десятков млн операторов!), появилось разделение труда: математики-алгоритмисты и программисты-тестировщики (или специалисты по качеству)

А. Математики-алгоритмисты:

- 1) формулировал постановку задачи;
- 2) разрабатывал алгоритм решения;
- 3) записывал алгоритм на используемом на данной машине ЯП;

Б. Программисты-тестировщики:

- 4) производил отладку и тестирование программы;
- 5) эксплуатировал разработанную программу;
- 6) при необходимости обновлял (вносил изменения) в программу;
- 7) прекращал пользоваться программой (время эксплуатации программы, как правило, недолгое, исчислялось месяцами, годом использования)

1. Разделение труда при разработке программ на ЯП (исторический взгляд)

III этап: середина 1970-х (ООП: Object Pascal, C++, Java, C# и мн. др.) – по настоящее время

Современные программы:

1) состоят из модулей, взаимодействующих между собой посредством десятков тысяч параметров;

2) трудоемкость написания программ составляет сотни и тысячи человеко-лет;

3) размер программ (в частности, ОС) – до миллиона операторов;

4) сложность написания программного обеспечения;

□ Программы стали эксплуатироваться на протяжении от нескольких лет до десятков лет

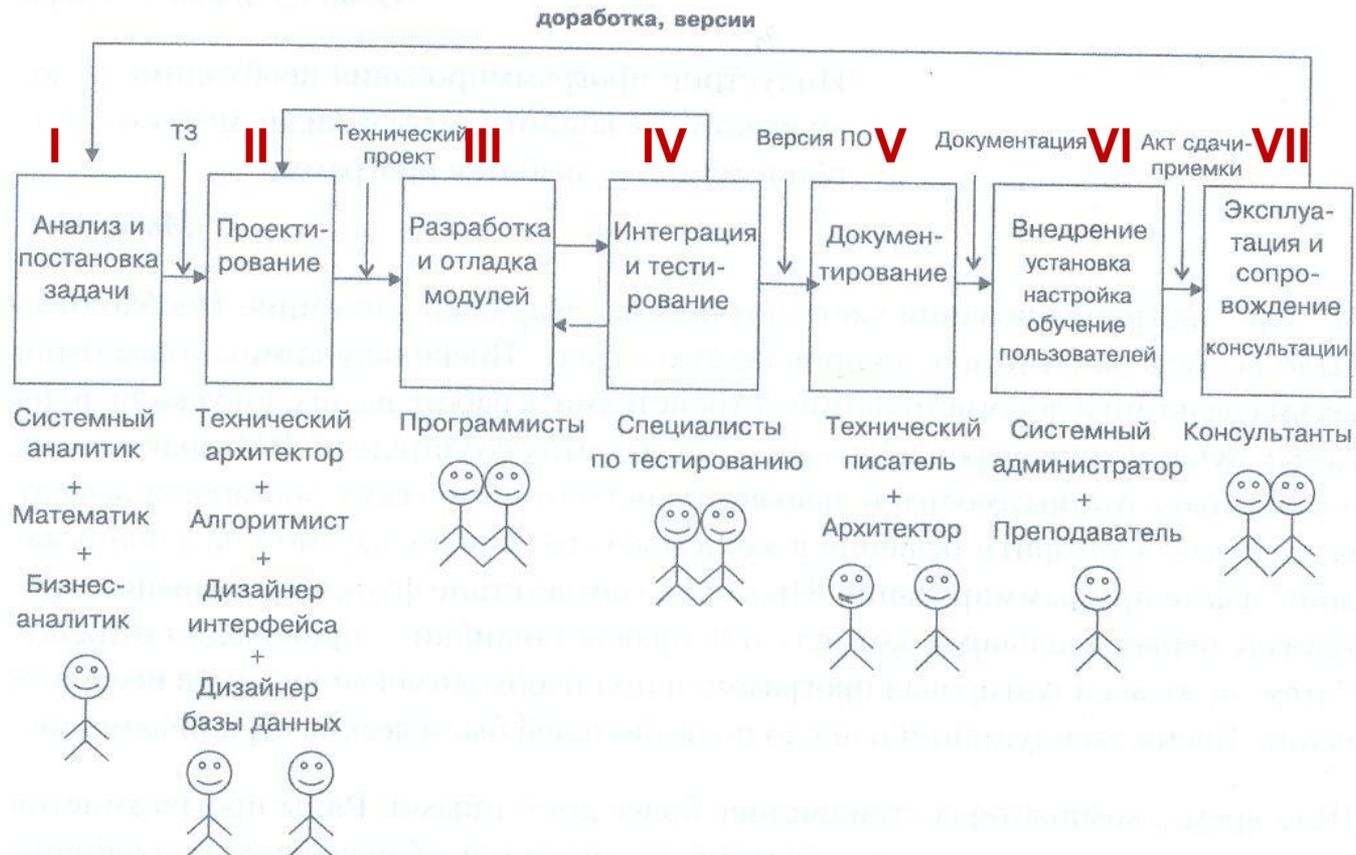
 Родители – «Жизненный цикл программной системы»

2. Понятие «Жизненный цикл программной системы»

Знать к экзамену(!)

Жизненный цикл программной системы –

– это все технологические этапы разработки программной системы, начиная от постановки задачи до вывода ее из эксплуатации



3. Основные этапы жизненного цикла программной системы

I. Постановка задачи и ее анализ –

- определение потребности в программной системе, оценка решаемости данной задачи (существует ли алгоритмическое решение);
- сколько времени понадобится для ее реализации;
- разработка технического задания системным аналитиком (четко сформулировать задачи!)

II. Проектирование программной системы –

- из каких подсистем (модулей) будет состоять программная система;
- какие языки программирования и платформы (ОС) будут использованы;
- Какие алгоритмы будут применяться;
- Как будут организованы входные данные (структуры данных);
- Как организовать интерфейс с пользователем программной системы и другим ПО.

III. Разработка и отладка модулей программы –

- выдача заданий программистам, отвечающим за свой модуль;
- тестирование и отладка (автономная отладка \equiv модульная отладка)

IV. Интеграция (сборка) и тестирование –

- сборка версии программного обеспечения и полное тестирование; (написание прогр.системы – 10%; отладка и тестирование – 55-60%)

3. Основные этапы жизненного цикла программной системы»

V. Документирование программной системы –

- выпуск официальной версии программной системы;
- документирование (руководства пишет «технический писатель»):
 - руководства пользователя;
 - руководства программиста;
 - руководства по установке и настройке программной системы;
 - методика испытаний;
 - учебник по программной системе.

VI. Внедрение: установка, настройка и обучение пользователей –

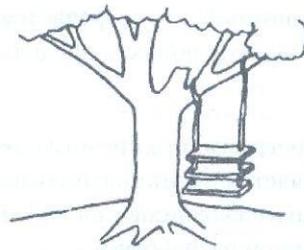
- передача заказчику программного продукта (акт сдачи-приемки);
- установка и настройка программной системы (сис.админ.);
- обучение работе с программой сотрудников.

VII. Эксплуатация и сопровождение программной системы –

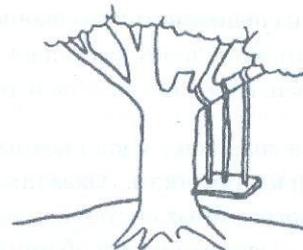
- организация консультаций пользователей;
- в случае выявления ошибок в процессе эксплуатации – исправление ошибок в программе (лучше, чтобы это происходило на этапе автономного тестирования);
- Современные программы эксплуатируются до нескольких лет (бывает десятков лет) и возникает потребность в доработке, внесении изменений, адаптации к меняющимся условиям производства и т.д. В этом случае выпускается частное техническое задание, которое согласовывают с заказчиком (возврат к I этапу – постановке задачи)

Вывод :

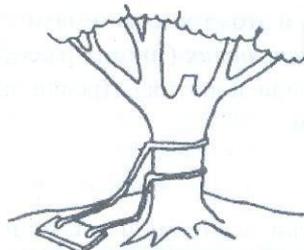
Чем точнее будут сформулированы задачи в техническом задании, тем меньше вероятность возникновения ситуации, когда выполнив большой объем работы, получают не совсем то, что было нужно (или совсем не то).



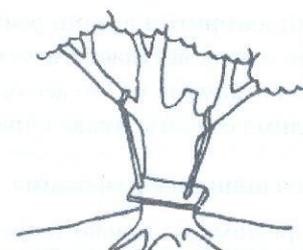
Как было предложено организатором разработки



Как было описано в техническом задании



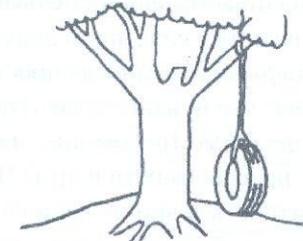
Как было спроектировано ведущим системным специалистом



Как было реализовано программистами



Как было внедрено



Что хотел пользователь