



IT

TELKOM-3

ЖИЗЕННЫЙ ЦИКЛ

ПО

«ЭКСПРЕСС-АМ5»

Следующая лекция из серии IT посвящена жизненному циклу программного продукта.

Концепция жизненного цикла является практической реализацией основных положений структурного программирования.

Программной инженерией называют инженерно-научную дисциплину, предметом которой является то, что должны знать и уметь лица, занимающиеся различными видами инженерной деятельности по созданию и использованию ПО, для профессионально качественного и высокопроизводительного выполнения своей работы.

Жизненный цикл ПО

В основе программной инженерии лежит подход к созданию программного обеспечения как к товарному продукту. Для этого применяются как административные методы, так и технология структурного программирования, но не менее важным является человеческий фактор, подбор, обучение кадров и привитие навыков работы в коллективе.

Начало бурному развитию **структурного программирования** как нового направления положила работа Дейкстры «GOTO считается вредным»* (1968 г.). Первоначально к нему относили все, что ускоряло разработку ПО, делало его понятнее и надежнее.

Сейчас в это направление принято включать концепции

- структурного кодирования,
- модульного программирования и
- дисциплинированного проектирования ПО.

Структурное кодирование составляют правила организации в программах связей по передачам управления — допускаются последовательности операторов, вызовы подпрограмм, цикла и ветвления, причем для всех управляющих конструкций допускаются варианты, имеющие только один вход и выход.

Современная концепция **модульного программирования** включает в себя следующие положения:

- программа разбивается на модули величиной 1-2 страницы распечатки (желательно);
- каждый модуль состоит из спецификаций и тела;
- должны иметься средства управления видимостью имен, ограничивающие множества доступных в каждом модуле внешних объектов;
- средства инструментальной поддержки должны обеспечивать возможность компиляции модулей до разработки тел используемых ими модулей.

Жизненный цикл ПО

Если предыдущие концепции рассматривали соответственно микро- и макроуровни ПО, то ***дисциплинарное проектирование*** относится к его развитию во времени. Процесс разработки ПО можно представить как последовательность актов принятия проектных решений с возвратами для исправления ошибочных решений. Проблема заключается в том, что принятие ошибочных решений в программировании не исключение, а скорее нормальный случай, причем обнаруживаются ошибки далеко не сразу.

Абстрактные типы данных

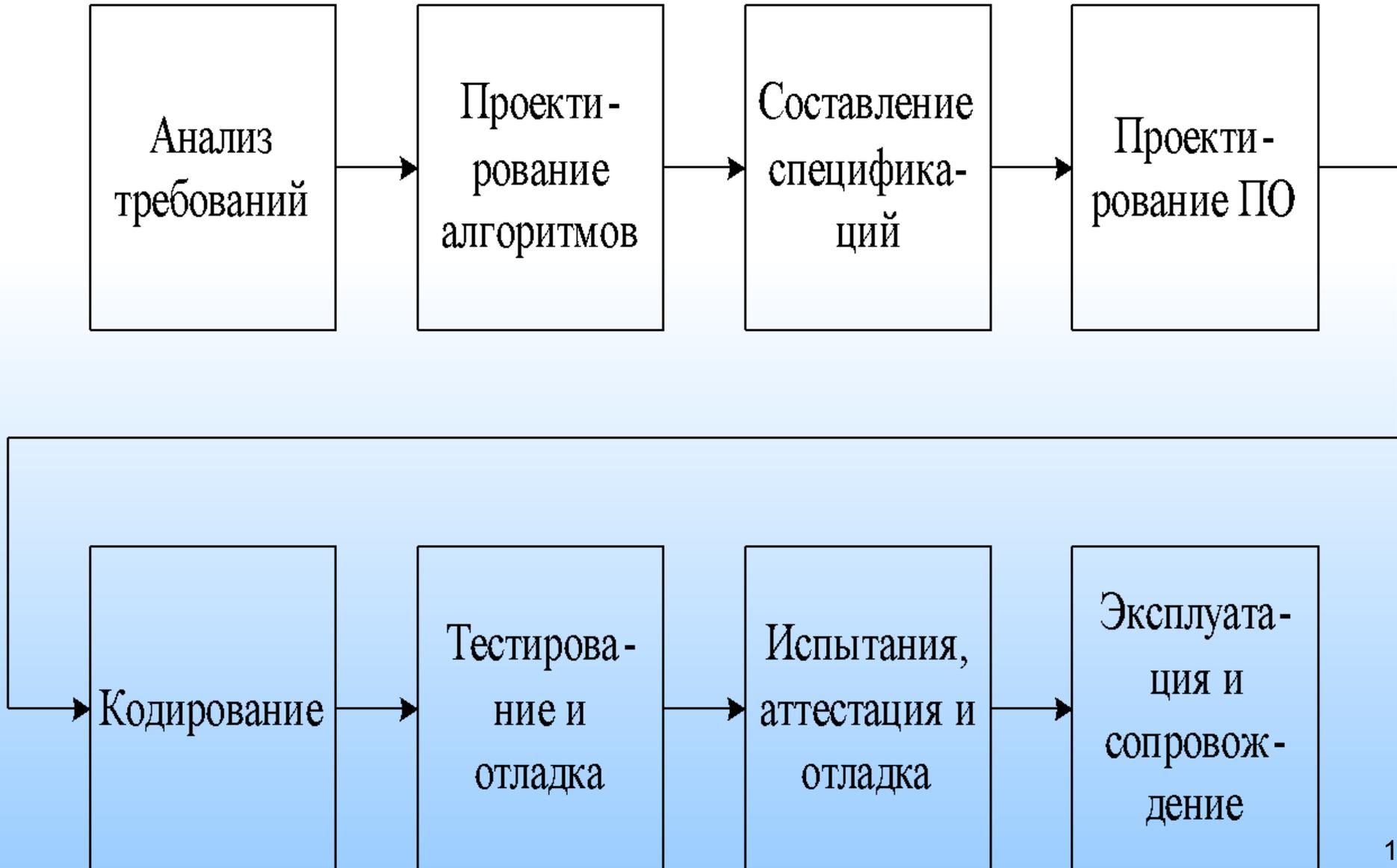
Структурное кодирование, аппарат подпрограмм и модульное программирование обеспечивают упорядочение в программах связей по передачам управления и накопление опыта в форме алгоритмов вычислений.

Поскольку основная сложность программ сосредоточена в структурах данных и связях по данным, необходимо и для данных иметь аналогичные средства.

Такая интерпретация типов данных была реализована Виртом в языке ПАСКАЛЬ. Она обеспечила абстрагирование от специфики конкретных ЭВМ, предоставила богатый выбор готовых типов данных и стандартных конструкторов типов данных (массивы, записи и т.д.), полезных для решения задач.

Далее будет рассмотрена современная технология проектирования программного продукта в виде жизненного цикла этого продукта.

Жизненный цикл ПО



История развития производства ПО

Жизненный цикл ПО состоит из восьми этапов как было показано на рисунке.

Показанная последовательность этапов отражает эволюционную модель. Реально, приходится многократно возвращаться к предыдущим этапам.

Далее это будет более понятно.

На ***этапе анализа требований***

уточняются цели проектирования; определяются частные цели и ресурсы проектирования, т. е. количество и квалификация специалистов, сроки и средства, выделенные на проектирование ПО, технические и программные средства технологического обеспечения этапов проектирования, отладки и сопровождения. Осуществляется выбор типа УВМ; определяется перечень задач, подлежащих решению. Работы на этом этапе, как правило, выполняются руководством без привлечения программистов.

Жизненный цикл ПО

На **этапе проектирования** алгоритма: выбираются математические методы решения задач в различных режимах обеспечивающие требуемую точность при заданных погрешностях исходных величин. Определяются все входные переменные для каждой задачи, диапазоны и скорости их изменения, массивы и типы констант, промежуточных и конечных результатов; составляются схемы информационных потоков и взаимных связей между задачами; задачи делятся на автономные модули.

Жизненный цикл ПО

Проектирование алгоритмов сопровождается моделированием алгоритмов на ЭВМ общего назначения с определением временных затрат на их исполнение в УВМ. По результатам моделирования уточняются требования к производительности и объемам памяти УВМ.

На ***этапе составления спецификаций*** определяются функции, выполняемые отдельными модулями; взаимные связи с другими модулями, последовательность их подключения.

На **этапе проектирования** ПО подробно разрабатывается внутренняя структура каждого модуля, правила и процедуры их взаимной связи, порядок и приоритетность выполнения; определяются информационные и управляющие связи, структуры данных, размещение модулей и данных в постоянной и оперативной памяти.

Жизненный цикл ПО

На этом этапе алгоритмы решения могут быть представлены на неформальном блочно-структурном языке, допускающем отображение производимых действий на естественный язык и в то же время позволяющем сравнительно легко осуществлять кодирование алгоритмов отдельных модулей на выбранном языке программирования (Ассемблер или некоторый проблемно – ориентированный язык).

Этап кодирования не занимает вопреки довольно распространенному мнению большой объем в распределении трудозатрат по жизненному циклу, которые зависят в общем случае от уровня применяемого языка программирования (чем выше уровень языка, тем меньше трудозатраты). На этом этапе осуществляется также комплексирование программ, т. е. объединение модулей, сопровождаемое редактированием их внутренних и внешних связей.

Под **тестированием** понимается проверка работоспособности программы, ее соответствие внешним спецификациям, а под **отладкой** - устранение выявленных в программе ошибок. Обе фазы тесно связаны между собой и взаимно дополняют друг друга.

Этап тестирования и отладки

распадается на ряд подэтапов: автономной статической отладки и тестирования, комплексной статической отладки и тестирования, динамической комплексной отладки и тестирования.

Автономной статической отладке и тестированию подвергаются отдельные программные модули. Как правило, автономное тестирование и отладка проводятся на ЭВМ общего назначения, снабженной интерпретатором команд объектной ЭВМ.

Комплексная статическая отладка и тестирование проводятся в основном также на технологической ЭВМ общего назначения для скомплексированных программных модулей с их внутренними и внешними связями. Они проводятся и на УВМ с комплексом программ, загруженным в отладочную память (имитатор памяти УВМ).

Наиболее сложными являются проведение ***динамической комплексной отладки и тестирования***, которые осуществляются на реальной УВМ с имитацией внешней среды на технологической ЭВМ, либо с подключением блоков и устройств реальной аппаратуры. Во всех перечисленных ситуациях решается проблема автоматической генерации тестов, массивов исходных данных, контрольных статических и динамических задач, фиксации результатов работы отлаживаемого комплекса программ.

Жизненный цикл ПО

На каждом из рассмотренных этапов жизненного цикла ПО работы сопровождаются выпуском документации (автоматизированным), однако она носит технологический характер и является, скорее, подсобным средством для программиста. Сюда относятся схемы алгоритмов и программ, листинги с комментариями программиста, листинги изменений, которые необходимо внести в программы, и др.

Жизненный цикл ПО

После проведения этапа тестирования и отладки, выпускается комплект документации, согласованный по объему и форме с заказчиком ПО, необходимый, с одной стороны, для обеспечения проведения испытаний и окончательной аттестации, а с другой - для изготовления штатных ПЗУ.

Жизненный цикл ПО

Завершающим этапом разработки является ***проведение испытаний ПО***, изготовленного на принятом носителе информации по документации разработчика и включенного в УВМ.

Эти испытания совмещаются, как правило, с испытаниями комплекса в целом, осуществляемыми в реальных условиях его функционирования. Наибольшие трудности здесь связаны с недостаточным совершенством или нехваткой специальных аппаратных и программных средств для обнаружения и фиксации появляющихся ошибок в работе ПО.

Жизненный цикл ПО

Для этих целей часто специально разрабатывается телеметрическая аппаратура, а также аппаратура документирования. После испытаний вносятся необходимые изменения в ПО и в документацию, затем ПО в составе серийных образцов УЭВМ поступает в эксплуатацию.

На **этапе эксплуатации и сопровождения ПО** отторгается от разработчика, и фиксация ошибок в ПО осуществляется в процессе его реального функционирования в организациях заказчика. Однако устранение выявленных погрешностей в работе ПО производится разработчиком. Им вносятся необходимые изменения в эксплуатационную и рабочую документацию, а также производится доработка выпущенных ранее изделий с данным комплектом ПО.

Жизненный цикл ПО

В организации и проведении этих работ и заключается сопровождение ПО. Чем больше объем ПО, тем сложнее поиск допущенных на предыдущих этапах программных ошибок, тем дороже обходится их исправление. Опыт эксплуатации сложных современных программных комплексов показывает, что трудозатраты при сопровождении ПО достигают 60—70 % от всего объема трудозатрат жизненного цикла ПО.



**Спасибо за
внимание!**