**Жизненный цикл программы**

Жизненный цикл начинается с потребности пользователя или создания программистом условий для потребности пользователя.

1. Обсуждение задания с заказчиком, уточнение технического задания на языке, понятном заказчику. Заключается письменный договор о ТЗ. На этом этапе выгодно просчитать, чтобы программа работала на новом железе и могла расширяться. Переговоры могут занимать от 15 минут до года.
2. Этап создания алгоритма, проектирование структур данных. Результат — алгоритм (схема или словесное описание, плюс описание структур данных). На этом этапе определяется язык программирования (язык должен поддерживаться и сейчас и через 5 лет) Перед описанием алгоритма, нужно поискать в уже существующих алгоритмах, тот, который подойдет нам. Если есть подходящие уже существующие алгоритмы — их надо использовать.
3. Кодировщик пишет код по готовому алгоритму.
4. Тестирование программы (демонстрация работы) Обычно компилятором выводится не более 1001 ошибки. Тестирование проводится при разных исходных данных, так чтобы проверить все ответвления. Исходные данные готовят программисты. Необходимо запланировать эксперимент, исходные данные для него. Ни одно тестирование не может гарантировать полное отсутствие ошибок в алгоритме. Нужно продемонстрировать тестирование программы заказчику, чтобы он подписал документ о приеме программы.
5. Сопровождение. Техническая документация позволяет через некоторое время вернуться к работе над программой (или начать работу над проектом новым программистам) (можно для этой цели использовать комментарии). Без технической документации и инструкции пользователю программа не считается законченной. Плановые и внеплановые изменения — часть жизненного цикла программы.
6. Программа устарела. Нужно написать новую программу, так, чтобы изменения были как можно менее фатальны для пользователя. В зависимости от управляемого процесса, процесс управления нельзя останавливать на определенное процессом время, следовательно, надо быстро переключать управляющие программы.
7. Жизненный цикл программы заканчивается, когда новая программа полностью берет на себя управление.

Развитие программного обеспечения идет по спирали!

Компиляция программы

Интерпретаторы (Basic) переводят строку за строкой.

Трансляторы проверяют так же строку за строкой и обнаруживают ошибки, совершенные ранее, там, где эта ошибка проявилась.

Трансляторы работают медленнее, однако на выходе дают объектный (исполняемый) файл.

Загрузчик обнаруживает свободное место в памяти, так, чтобы переменная там поместилась, и размещает.

Ассемблер — программа, которая собирает в кучу все пакеты.

Компилятор собирает все пакеты сразу при компиляции.

После ассемблирования загрузчик знает адреса и кидает их в объектный файл.

Как правило элементы размещаются в памяти как можно плотнее, при этом увеличивается скорость загрузки!

В результате получается исполняемый файл .exe

Когда получен исполняемый файл, исполняемая программа уже не меняется (физически).