

Лекция 1. Основы программной инженерии

Основой проектирования программного обеспечения является системный подход. *Системный подход* – это методология исследования объекта любой природы как системы. *Система* – это совокупность взаимосвязанных частей, работающих совместно для достижения некоторого результата. Определяющий признак системы – поведение системы в целом не сводимо к совокупности поведения частей системы.

Программное обеспечение – это система, включающая в себя: компьютерные программы; документацию; данные, необходимые для корректной работы программ.

Проектирование ПО – это процесс создания спецификаций ПО на основе исходных требований к нему.

Проект ПО – совокупность спецификаций ПО (включающих модели и проектную документацию), обеспечивающих создание ПО в конкретной программно-технической среде.

ПО можно разбить на два класса: «малое» и «большое».

«Малое» программное обеспечение имеет следующие характеристики:

- решает одну несложную, четко поставленную задачу;
- размер исходного кода не превышает нескольких сотен строк;
- скорость работы программного обеспечения и необходимые ему ресурсы не играют большой роли;
- ущерб от неправильной работы не имеет большого значения;
- модернизация программного обеспечения, дополнение его возможностей требуется редко;
- как правило, разрабатывается одним программистом или небольшой группой (5 или менее человек);
- подробная документация не требуется, ее может заменить исходный код, который доступен.

«Большое» программное обеспечение имеет 2-3 или более характеристик из следующего перечня:

- решает совокупность взаимосвязанных задач;
- использование приносит значимую выгоду;
- удобство его использования играет важную роль;
- обязательно наличие полной и понятной документации;
- низкая скорость работы приводит к потерям;
- сбои, неправильная работа, наносит ощутимый ущерб;
- программы в составе ПО во время работы взаимодействуют с другими программами и программно-аппаратными комплексами;
- работает на разных платформах;
- требуется развитие, исправление ошибок, добавление новых возможностей;
- группа разработчиков состоит из более 5 человек.

Далее рассматривается проектирование «большого» ПО, поскольку создание «малого» не вызывает больших трудностей, не требует специальной технологии и инструментов.

Классификация программных проектов по размеру группы разработчиков и длительности проекта:

- *небольшие проекты* – проектная команда менее 10 человек, срок от 3 до 6 месяцев;
- *средние проекты* – проектная команда от 20 до 30 человек, протяженность проекта 1-2 года;
- *крупномасштабные проекты* – проектная команда от 100 до 300 человек, протяженность проекта 3-5 лет;
- *гигантские проекты* – армия разработчиков от 1000 до 2000 человек и более (включая консультантов и соисполнителей), протяженность проекта от 7 до 10 лет.

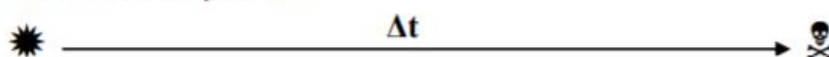
Особенности современных проектов ПО:

- сложность – неотъемлемая характеристика создаваемого ПО;
- отсутствие полных аналогов и высокая доля вновь разрабатываемого ПО;
- наличие унаследованного ПО и необходимость его интеграции с разрабатываемым ПО;
- территориально распределенная и неоднородная среда функционирования;
- большое количество участников проектирования, разобщенность и разнородность отдельных групп разработчиков по уровню квалификации и опыту.

Разработка ПО имеет следующие специфические особенности:

- неформальный характер требований к ПО и формализованный основной объект разработки – программы;
- творческий характер разработки;
- дуализм ПО, которое, с одной стороны, является статическим объектом – совокупностью текстов, с другой стороны, – динамическим, поскольку при эксплуатации порождаются процессы обработки данных;
- при своем использовании (эксплуатации) ПО не расходуется и не изнашивается;

Основным понятием программной инженерии является понятие *жизненного цикла ПО*. *Жизненный цикл ПО (software lifecycle)* – это период времени, который начинается с момента принятия решения о необходимости создания ПО и заканчивается в момент его полного изъятия из эксплуатации.



Основной нормативный документ, регламентирующий ЖЦ ПО – стандарт ISO/IEC 12207 “Information Technology – Software Life Cycle Processes” (ГОСТ Р ИСО/МЭК 12207-99). В рамках технологий создания ПО понятие ЖЦ уточняется, но указанные стандарты не нарушаются. Процесс стандартизации ведётся довольно интенсивно. Стандарт ISO/IEC 12207 имеет несколько редакций с 1995 по 2008 год.

С точки зрения статической структуры ЖЦ является совокупностью процессов ЖЦ.

Процесс ЖЦ – набор взаимосвязанных действий, преобразующих некоторые входные данные и ресурсы в выходные.

Каждый процесс характеризуется задачами, методами их решения, действующими лицами, результатами. Процессы ЖЦ протекают параллельно. Каждый процесс разделен на набор действий, каждое действие – на набор задач. Каждый процесс, действие или задача инициируется и выполняется по мере необходимости, причем не существует заранее определенных последовательностей выполнения. Группы процессов ЖЦ (ISO/IEC 12207:1995):

- основные (приобретение, поставка, разработка, эксплуатация, сопровождение);
- вспомогательные (документирование, управление конфигурацией, обеспечение качества, верификация, аттестация, совместная оценка, аудит, разрешение проблем);
- организационные (управление, создание инфраструктуры, усовершенствование, обучение).

Для ознакомления приведем содержание процессов ЖЦ.

Процесс приобретения включает следующие действия: инициирование приобретения; подготовку заявочных предложений; подготовку и корректировку договора; надзор за деятельностью поставщика; приемку и завершение работ. Действующие лица: заказчик, поставщик. Задачи приобретения: определение заказчиком своих потребностей в ПО; анализ требований к ПО; принятие решения о приобретении ПО; выработка плана приобретения и заявочных предложений; выбор поставщика; подготовка и заключение договора с поставщиком; контроль за соблюдением условий договора; корректировка договора при необходимости.

Процесс поставки включает в себя следующие действия: инициирование поставки; подготовку ответа на заявочные предложения; подготовку договора; планирование и выполнение поставки; контроль поставки; проверку и оценку. Действующие лица: заказчик, поставщик. Задачи поставки: оценка поставщиком заявочных предложений; подготовка и заключение договора с заказчиком, контроль со стороны поставщика за соблюдением условий договора, принятие решения о привлечении субподрядчика или выполнении работ своими силами, выработка плана управления проектом и др.

Процесс разработки включает в себя следующие действия: подготовительную работу; анализ требований к ПО; проектирование архитектуры ПО; детальное проектирование ПО; кодирование ПО; тестирование ПО; интеграцию ПО; установку ПО; приемку ПО. Действующие лица: разработчик, заказчик. Задачи разработки: выбор модели ЖЦ ПО и согласование с заказчиком; определение требований к ПО (функциональных и нефункциональных); определение состава компонентов ПО и создание документации по каждому компоненту; моделирование и спецификация компонент ПО; планирование интеграции компонент; создание исходных текстов компонент; поиск и исправление ошибок в исходных текстах и документации; сборка ПО; развертывание ПО; оценка результатов.

Процесс эксплуатации включает в себя следующие действия: подготовительную работу; эксплуатационное тестирование; эксплуатацию; поддержку пользователей. Действующие лица: оператор (организация, эксплуатирующая ПО), пользователи. Задачи эксплуатации: выработка плана эксплуатации и эксплуатационных стандартов; составление процедур локализации и разрешения проблем эксплуатации; поиск ошибок в ПО перед вводом в эксплуатацию его новых версий; оказание помощи пользователям и консультирование.

Процесс сопровождения включает в себя следующие действия: подготовительную работу; анализ проблем и запросов на модификацию ПО; проверку и приемку; перенос ПО в другую среду; снятие ПО с эксплуатации. Действующие лица: служба сопровождения, пользователи. Задачи сопровождения: выработка плана сопровождения; составление процедур локализации и разрешения проблем сопровождения; оценка целесообразности внесения модификаций в ПО; принятие решения о модификации; поиск ошибок в ПО после его модификации; проверка целостности ПО; архивирование при снятии с эксплуатации; обучение пользователей.

Процесс документирования включает в себя следующие действия: подготовительную работу; проектирование и разработку документации; выпуск документации; сопровождение.

Процесс управления конфигурацией включает в себя следующие действия: подготовительную работу; создание базы знаний о ПО (конфигурации); контроль за конфигурацией; учет состояния конфигурации; оценку конфигурации; управление выпуском и поставку ПО. *Конфигурация ПО* – это совокупность сведений о его функциональных и физических характеристиках на всех стадиях ЖЦ ПО. Основная задача управления конфигурацией: организация, систематический учет и контроль внесения изменений в ПО.

Процесс обеспечения качества включает в себя следующие действия: подготовительную работу; обеспечение качества продукта; обеспечение качества процесса; обеспечение других показателей качества ПО. Задачи обеспечения качества: гарантированное соответствие ПО требованиям заказчика, зафиксированным в договоре; гарантированное соответствие процессов ЖЦ ПО, методов разработки, квалификации персонала установленным стандартам.

Процесс верификации включает в себя следующие действия: подготовительную работу; верификацию. Основная задача верификации – проверка соответствия разработанных программ в составе ПО их спецификациям.

Процесс аттестации состоит в определении полноты соответствия разработанного ПО требованиям заказчика. Основная задача аттестации – оценка достоверности тестирования ПО. Как правило, для аттестации привлекают независимых экспертов.

Процесс совместной оценки включает в себя следующие действия: подготовительную работу; оценку управления проектом; техническую оценку. Основная задача совместной оценки – контроль планирования и управления ресурсами, персоналом, инфраструктурой проекта.

Процесс аудита состоит в определении полноты соответствия проекта условиям договора.

Процесс разрешения проблем предусматривает анализ и разрешение проблем, возникающих в течение ЖЦ ПО.

Процесс управления включает в себя следующие действия: подготовительную работу; планирование; выполнение и контроль; проверку и оценку; завершение. Задачи управления: проверка достаточности имеющихся ресурсов; составление графиков работ; оценка затрат; выделение ресурсов; распределение ответственности; оценка рисков.

Процесс создания инфраструктуры состоит в выборе и поддержке технологии разработки ПО, стандартов и инструментальных средств; выборе и установке аппаратных и программных средств, необходимых для разработки, эксплуатации и сопровождения ПО.

Процесс усовершенствования предусматривает оценку, измерение, контроль и усовершенствование процессов ЖЦ ПО. Основная задача усовершенствования – повышение производительности труда.

Процесс обучения включает в себя следующие действия: подготовительную работу; разработку учебных планов, курсов, материалов; реализацию планов обучения. Задачи обучения: первоначальное обучение персонала; повышение квалификации персонала.

В обновленной версии ISO/IEC 12207:2008 рассматриваются больше процессов ЖЦ:

Группа процессов договора: приобретение и поставка.

Группа организационных процессов: управление моделью ЖЦ, управление инфраструктурой, управление проектным портфолио, управление кадрами, управление качеством.

Группа технических процессов: определение интересов участников, анализ требований к системе, проектирование системной архитектуры, реализация, интеграция системы, аттестация, установка, приёмка, эксплуатация системы, сопровождение, вывод из эксплуатации.

Группа процессов ПО (software-specific processes) включает три подгруппы: процессы создания ПО (группа из 7 пр.); процессы поддержки ПО (8 процессов); процессы повторного использования ПО (3 процесса).

Сравнение номенклатуры процессов ЖЦ в двух версиях стандарта ISO/IEC 12207 отражает происходящее в программной инженерии накопление опыта.

Вторым измерением ЖЦ, дополняющим структурное, но ортогональным ему, является *динамическое*, определяющее развитие ЖЦ во времени в виде *модели жизненного цикла*.

Модель ЖЦ ПО – это структура, определяющая последовательность выполнения и взаимосвязи процессов, действий и задач на протяжении всего ЖЦ.

В любой модели ЖЦ рассматривается как совокупность стадий ЖЦ.

Стадия ЖЦ – это часть ЖЦ ограниченная временными рамками, по завершении которой достигается определенный важный результат в соответствии с требованиями для данной стадии ЖЦ. Между двумя стадиями, идущими друг за другом, находится *контрольная точка (vexa)*. Так называют момент времени, разделяющий стадии жизненного цикла (или итерации, если они предусмотрены в модели ЖЦ), по наступлении которого должны достигаться результаты важные для всего проекта и должны приниматься решения о дальнейшей разработке.



Модели ЖЦ:

- каскадная (водопадная);
- эволюционная;
- основанная на формальных преобразованиях;
- итерационные (пошаговая и спиральная).

Схема каскадной модели ЖЦ:

Принципы *каскадной модели*: фиксация требований к системе в начале проекта; переход со стадии на стадию только после полного завершения работ на текущей стадии; жесткая привязка процессов ЖЦ к стадиям ЖЦ; недопустимость возврата на поздних стадиях к работам, приписанным к пройденным стадиям.

Стадия *формирования требований* включает процессы, приводящие к созданию документа, описывающего поведение ПО с точки зрения внешнего по отношению к нему наблюдателя с фиксацией требований

относительно его качества.

Проектирование охватывает процессы: разработку архитектуры ПО, разработку структур программ в его составе и их детальную спецификацию.

Реализация или кодирование включает процессы создания текстов программ на языках программирования.

На этапе *тестирования* производится собственно тестирование, а также отладка и оценка качества ПО.

Ввод в действие – это развертывание ПО на целевой вычислительной системе, обучение пользователей и т.п.

Эксплуатация ПО – это использование ПО для решения практических задач на компьютере путем выполнения ее программ.

Сопровождение ПО – это охватывает сбор информации о качестве ПО в эксплуатации, устранение обнаруженных в нем ошибок, его доработку и модификацию, а также извещение пользователей о внесенных в него изменениях.

Обратите внимание на схожесть списка стадий в водопадной модели с перечнем технических процессов в ISO/IEC 12207:2008.

Достоинства: на каждой стадии формируется законченный набор проектной документации, отвечающий критериям полноты и согласованности; выполняемые в логичной последовательности стадии работ облегчают планирование сроков завершения всех работ и соответствующих затрат. Недостатки: позднее обнаружение проблем; выход из календарного графика, запаздывание с получением результатов; высокий риск создания системы, не удовлетворяющей изменившимся потребностям пользователей; избыточность документации; неравномерная нагрузка членов группы, работающей над проектом в ходе ЖЦ.



Неизбежные возвраты к работам, приписанным к предыдущим стадиям, в каскадной модели

На самом деле невозможно двигаться строго поступательно, необходимо возвращаться – пересматривать требования, перепроектировать, изменять реализацию и т. д., чтобы исправлять ошибки, сделанные на ранних стадиях, устранять недостатки, учитывать меняющиеся в ходе проекта требования. В этом кроется причина недостатков водопадной модели. Недостатки не вычёркивают водопадную модель из списка применимых на практике. Для любой модели ЖЦ они должны рассматриваться как продолжения достоинств.

