

Основные понятия технологии программирования



Лекция 1

Литература

1. Орлов С.А., Цилькер Б.Я. Технологии разработки программного обеспечения. Современный курс по программной инженерии: Учебник для вузов. 4-е изд. – СПб., Питер, 2012. – 608 с.: ил.
2. Соммервилл И. Инженерия программного обеспечения.: Пер. с англ.: – М., Вильямс, 2002. – 623 с.: ил.
3. Брауде Э. Дж. Технология разработки программного обеспечения.: Пер. с англ.: – СПб., Питер, 2004.– 654 с.: ил.

Литература

4. Якобсон А., Буч Г., Рамбо Д. Унифицированный процесс разработки программного обеспечения.: Пер. с англ.: – СПб., Питер, 2002. – 492 с.: ил.
5. Жоголев Е. А. Технология программирования: М., Научный мир, 2004. – 215 с.: ил.
6. Терехов А.Н. Технология программирования: М., ИНТУИТ, 2006. – 152 с.: ил.

Литература

7. Гамма Э., Хелм Р., Джонсон Р., Влиссидес Д. Приемы объектно-ориентированного проектирования. Паттерны проектирования.: Пер. с англ.: – СПб., Питер-ДМК, 2001. – 366 с. ил.
8. В. В. Кулямин. Технологии программирования. Компонентный подход.
<http://panda.ispras.ru/~kuliainin/lectures-sdt/sdt-book-2006.pdf>

Программы «большие» и «маленькие»

- Основная тема данного курса — методы разработки *«больших» и сложных* программ
- Программы, решающие какие-либо узкие задачи, в большинстве случаев пишутся программистами «на коленках» без использования особых методов
- Пример: программа для вычисления значения синуса угла с некоторой наперед заданной точностью

Особенности «маленьких» программ

- Для «малых» программ можно указать следующие характерные особенности:
 - сравнительно небольшой размер (порядка нескольких сотен строк);
 - направленность на решение одной четко поставленной задачи с хорошо известными ограничениями;
 - отсутствие оптимизации по скорости выполнения;

Особенности «маленьких» программ

а также

- практическое отсутствие ущерба от неправильной работы программы;
- отсутствие необходимости дополнять программу новыми возможностями и разрабатывать ее новые версии;
- минимальная потребность в документировании

«Большие» программы

- «Большие» программы и программные комплексы создаются для решения сложных задач, связанных с практической деятельностью значительного числа людей
- Примерами таких программ являются всевозможные системы автоматизации производственных процессов, системы управления и контроля, СУБД и т.д

Свойства «больших» программ

- «Большая» программа обычно обладает следующими свойствами:
 - решает одну или несколько связанных задач, важных для каких-либо лиц или организаций, получающих существенные выгоды от ее использования;
 - ее неправильная работа или низкая производительность на реальных данных наносит ощутимый ущерб пользователям и другим организациям и лицам;

Свойства «больших» программ

- сопровождается полной и понятной пользователям документацией, а также специальной документацией для администраторов;
- дополняется проектной документацией к ПС, что позволяет развивать ее даже без привлечения первоначальных разработчиков;
- такую программу невозможно написать с первой попытки, с небольшими усилиями и в одиночку; ее разработка ведется коллективом специалистов

Программное обеспечение

- Как правило, «большие» программы требуют для своего выполнения некоторого набора аппаратных средств, образуя *программно-аппаратные системы*
- Поэтому иногда мы будем пользоваться понятием «программное обеспечение» («ПО»), подразумевая под этим собственно программную «начинку» программно-аппаратных систем

Программная инженерия

- *Программная инженерия (Software Engineering)* – это отрасль информатики, которая изучает вопросы построения компьютерных программ, закономерности развития программирования, а также обобщает опыт программирования в виде комплекса знаний и правил регламентации инженерной деятельности разработчиков ПО

Программная инженерия

- *Инженерия* - это способ применения научных результатов, для извлечения пользы от свойств материалов и источников энергии
- Инженеры в программной инженерии - это специалисты, выполняющие практические работы по реализации программ с применением теории, методов и средств компьютерной науки

Программная инженерия

- Компьютерная наука охватывает теорию и методы построения вычислительных и программных систем
- Программная инженерия рассматривает вопросы практического построения ПО
- Цель науки – получение знаний, для инженерии знание – это способ получения некоторой пользы

Виды деятельности

- Кроме программистов, занимающихся непосредственно разработкой ПО, деятельностью в сфере программной инженерии занимаются:
 - менеджеры, которые планируют и руководят проектом, отслеживают сроки и затраты;
 - инженеры службы ведения библиотек и репозитариев компонентов;

Виды деятельности

□ А также

- технологи, которые определяют инженерные методы и стандарты,;
- тестировщики, контролирующие правильность выполнения процесса разработки ПО;
- верификаторы, проверяющие ПО на соответствие заданным спецификациям;
- валидаторы, проверяющие ПО на соответствие заданным условиям эксплуатации.

Технология программирования

- Итогом инженерной деятельности в плане освоения достижений компьютерной науки и обобщения практического опыта программирования является *технология программирования* – совокупность принципов разработки, обеспечивающих массовое производство ПО требуемого качества в установленные сроки

Методы и средства ТП



Методы ТП

- *Методами* технологии программирования называются способы и приемы организации производственных процессов при разработке программных средств
- Методы ТП определяют организационную структуру коллектива разработчиков, способы разбиения процесса разработки на отдельные этапы, последовательность этих этапов и т.д.

Средства ТП

- Средствами технологии программирования называются утилиты, обеспечивающие автоматизированную или автоматическую поддержку методов
- Совместно используемые утилиты объединяются в системы автоматизированной разработки ПО
- Такие системы принято называть CASE-средствами (Computer Aided Software Engineering)

Цели ТП

- *Цели* технологии программирования сформулированы уже в ее определении – производство ПО требуемого качества в установленные сроки
- Очевидно, что
 - обеспечение требуемого качества связано со способами деятельности,
 - соблюдение сроков связано с организацией деятельности

Проблемы качества ПО

- К сожалению, положение дел с обеспечением качества ПО остается неудовлетворительным
- По данным Национального Института стандартов и технологий, ошибки в программном обеспечении обходятся экономике США в 60 млрд. долларов в год, а в мировом масштабе они, по крайней мере, втрое выше

Проблемы качества ПО

- Новый программный проект создается 1-2 года, а эволюционирует 6-7 лет
- На сопровождение проекта, включая его доработку и исправление ошибок, тратится 61% средств против 39% на его разработку

Проблемы качества ПО

- Наблюдаются две основные тенденции:
 - значительное увеличение производительности программистов в терминах количества кода, создаваемого ими в единицу времени;
 - сохранение среднего количества ошибок в пределах 10-50 на тысячу строк кода, еще не прошедшего тестирование

Почему это так?

- **Две основные причины:**
 - сложность современных программных комплексов такова, что многие исследователи считают их самыми сложными системами, созданными человеком, и поэтому практически невозможно выполнить полное доказательство или проверку их правильности;
 - требования к ПО у разных групп пользователей могут существенно различаться и значительно изменяться с течением времени

Понятие качества программного обеспечения

- *Качество ПО* – это вся совокупность его характеристик, относящихся к возможности удовлетворять высказанные или подразумеваемые потребности всех заинтересованных лиц

Международный стандарт

- Основой регламентирования показателей качества программных систем является международный стандарт ISO 9126 «Информационная технология. Оценка программного продукта. Характеристики качества и руководство по их применению»
- Стандарт определяет ряд *критериев качества* программного продукта

Критерии качества ПО

- Основными критериями качества ПО (*criteria of software quality*) являются:
 - функциональность
 - надежность
 - эффективность
 - эргономичность
 - модифицируемость
 - мобильность

Функциональность ПО

- Способность ПО выполнять набор функций (действий), удовлетворяющих заданным или подразумеваемым потребностям пользователей
- Набор указанных функций определяется во внешнем описании ПО

Надежность программного обеспечения

- *Надежность (reliability)* ПО – это его способность с достаточно большой вероятностью безотказно выполнять определенные функции при заданных условиях и в течение заданного периода времени

Эффективность программного обеспечения

- Соотношение уровня услуг, предоставляемых ПО пользователю при заданных условиях, и объема используемых для этого ресурсов
- К числу таких ресурсов могут относиться требуемые аппаратные средства, время выполнения программ, затраты на подготовку данных и интерпретацию результатов

Эргономичность ПО

- Характеристики ПО, которые позволяют минимизировать усилия пользователя по подготовке исходных данных, применению ПО и оценке полученных результатов, а также вызывать положительные эмоции определенного или подразумеваемого пользователя

Модифицируемость программного обеспечения

- Характеристики ПО, которые позволяют минимизировать усилия по внесению изменений для устранения ошибок и по его модификации в соответствии с изменяющимися потребностями пользователей
- Модифицируемость ПО существенно зависит от степени и качества его документированности

Мобильность ПО

- Способность ПО быть перенесенным из одной среды (окружения) в другую, в частности, с одной аппаратной платформы на другую

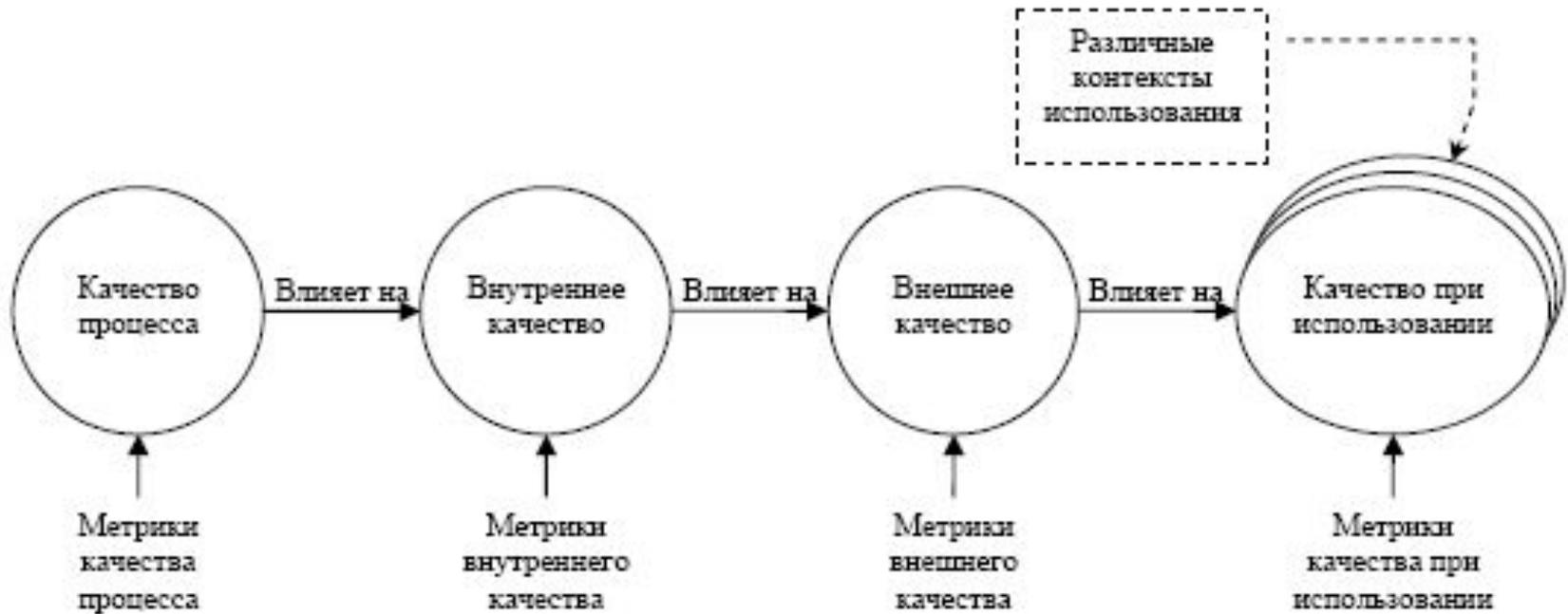
Стандарт ISO 9126

- Международный стандарт, определяющий оценочные характеристики качества программного обеспечения, утвержден в 1991 году
- Стандарт вводит понятия:
 - внутреннего качества,
 - внешнего качества,
 - качества ПО при использовании

Три аспекта качества ПО

- *Внутреннее качество* связано с характеристиками ПО самого по себе, без учета его поведения
- *Внешнее качество* характеризующего ПО с точки зрения его поведения
- *Качества ПО при использовании* – это то качество, которое ощущается пользователями при конкретных сценариях работы ПО

Три аспекта качества ПО



Структура стандарта ISO 9126

- Стандарт разделяется на 4 части, описывающие следующие вопросы:
 - модель качества;
 - внешние метрики качества;
 - внутренние метрики качества;
 - метрики качества в использовании

Модель качества

- Стандарт ISO 9126 предлагает использовать для описания внутреннего и внешнего качества ПО *многоуровневую модель*
- На верхнем уровне выделено 6 основных характеристик качества ПО. Каждая характеристика описывается при помощи нескольких входящих в нее *атрибутов*. Для каждого атрибута определяется набор метрик, позволяющих его оценить

Модель качества



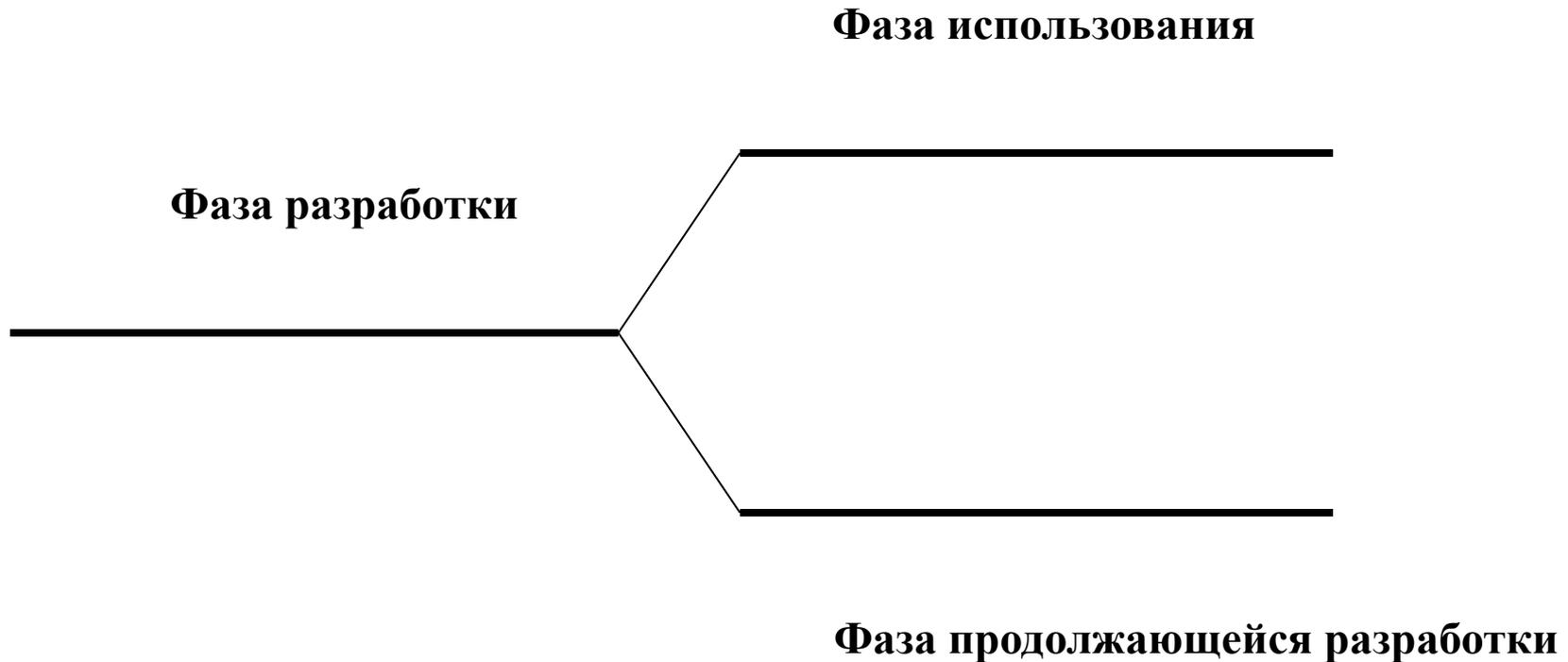
Проблемы разработки программного обеспечения

- Основные проблемы создания сложных программных систем связаны с нахождением разумного компромисса между затратами на разработку и качеством ее результата

Жизненный цикл ПО

- *Жизненным циклом* программного обеспечения называется весь период времени от начала его разработки до завершения использования
- Жизненный цикл ПО состоит из *фазы разработки, фазы использования и фазы продолжающейся разработки (модификации)*, причем две последние фазы близки или совпадают по времени

Фазы жизненного цикла



Этапы фазы разработки

- Наиболее интересной фазой жизненного цикла ПО является фаза разработки
- Эта фаза может быть разбита на ряд этапов, а именно:
 - анализ системы и выявление требований к ПО;
 - проектирование ПО;
 - конструирование (кодирование) ПО;
 - тестирование ПО;
 - инсталляция ПО

Артефакты

- Жизненный цикл ПО связан с различными видами деятельности большого количества людей
- При этом создаются и перерабатываются различного рода *артефакты* – создаваемые человеком информационные сущности (документы), участвующие в качестве входных данных и результатов в различных видах деятельности

Примеры артефактов

- Примерами артефактов являются:
 - модель предметной области,
 - описание требований,
 - техническое задание,
 - описание архитектуры системы,
 - проектная документация на систему в целом и на отдельные ее компоненты,
 - прототипы системы и компонентов,
 - ИСХОДНЫЙ КОД,

Примеры артефактов

- пользовательская документация,
- документация администратора системы,
- руководство по развертыванию,
- база пользовательских запросов,
- план проекта

Роли

- На различных этапах в создание и эксплуатацию ПО вовлекаются люди, выполняющие различные *роли*
- Каждая роль может быть охарактеризована как абстрактная группа заинтересованных лиц, участвующих в деятельности по созданию и эксплуатации системы и решающих одни и те же задачи или имеющих одни и те же интересы по отношению к ней

Примеры ролей

- Примерами ролей являются:
 - бизнес-аналитик,
 - инженер по требованиям,
 - архитектор,
 - проектировщик пользовательского интерфейса,
 - программист-кодировщик,
 - технический писатель,
 - тестировщик,

Примеры ролей

- руководитель проекта по разработке,
- работник отдела продаж,
- конечный пользователь,
- администратор системы,
- инженер по поддержке и т.п.

Стандарт ISO/IEC 12207-95

- По определению, ISO/IEC 12207-95 — базовый стандарт процессов ЖЦ ПО, ориентированный на различные виды ПО и типы проектов автоматизированных систем, куда ПО входит как часть
- Стандарт определяет стратегию и общий порядок в создании и эксплуатации ПО, он охватывает ЖЦ ПО от концептуализации идей до завершения ЖЦ

Стандарт ISO/IEC 12207-95

- Первая редакция ISO/IEC 12207-95 подготовлена в 1995 году объединенным техническим комитетом ISO/IEC JTC1 "Информационные технологии, подкомитет SC7, проектирование программного обеспечения»

Определения стандарта: модель ЖЦ

- *Модель жизненного цикла* — структура, содержащая процессы, действия и задачи, которые осуществляются в ходе разработки, функционирования и сопровождения программного продукта в течение всей жизни системы, от определения требований до завершения ее использования.

Модель ЖЦ

- Стандарт определяет общую структуру жизненного цикла ПО в виде трехступенчатой модели, состоящей из
 - процессов,
 - видов деятельности,
 - задач
- Стандарт не определяет метрики, по которым можно было бы отслеживать ход работ и их результативность

Процессы жизненного цикла

- Самыми крупными элементами являются *процессы жизненного цикла ПО*
- Всего выделено 18 процессов, которые объединены в 4 группы:
 - основные процессы,
 - поддерживающие процессы,
 - организационные процессы,
 - процесс адаптации

Процессы ЖЦ по ISO 12207

Основные процессы	Поддерживающие процессы	Организационные процессы	Адаптация
Приобретение ПО; Передача ПО (в использование); Разработка ПО; Эксплуатация ПО; Поддержка ПО	Документирование; Управление конфигурациями; Обеспечение качества; Верификация; Валидация; Совместные экспертизы; Аудит; Разрешение проблем	Управление проектом; Управление инфраструктурой; Усовершенствование процессов; Управление персоналом	Адаптация описываемых стандартом процессов под нужды конкретного проекта

Действия и задачи

- Каждый процесс ЖЦ разделен на *набор работ (activities)*, каждое действие — на *набор задач (tasks)*
- Всего определены 74 вида работ и 224 различных задач
- Каждый процесс, работа или задача инициируется и выполняется другим процессом по мере необходимости

Основные процессы ЖЦ

- *Процесс разработки.* Определяет действия предприятия-разработчика, которое создает программный продукт
- Включает следующие работы:
 - развертывание процесса разработки,
 - анализ системных требований,
 - проектирование (программно-аппаратной) системы в целом,

Основные процессы ЖЦ

- анализ требований к ПО,
- проектирование архитектуры ПО,
- детальное проектирование,
- кодирование
- отладочное тестирование,
- интеграцию ПО,
- квалификационное тестирование ПО,
- системную интеграцию

Основные процессы ЖЦ

- квалификационное тестирование системы,
- развертывание (установку или инсталляцию)
ПО

Конец лекции
