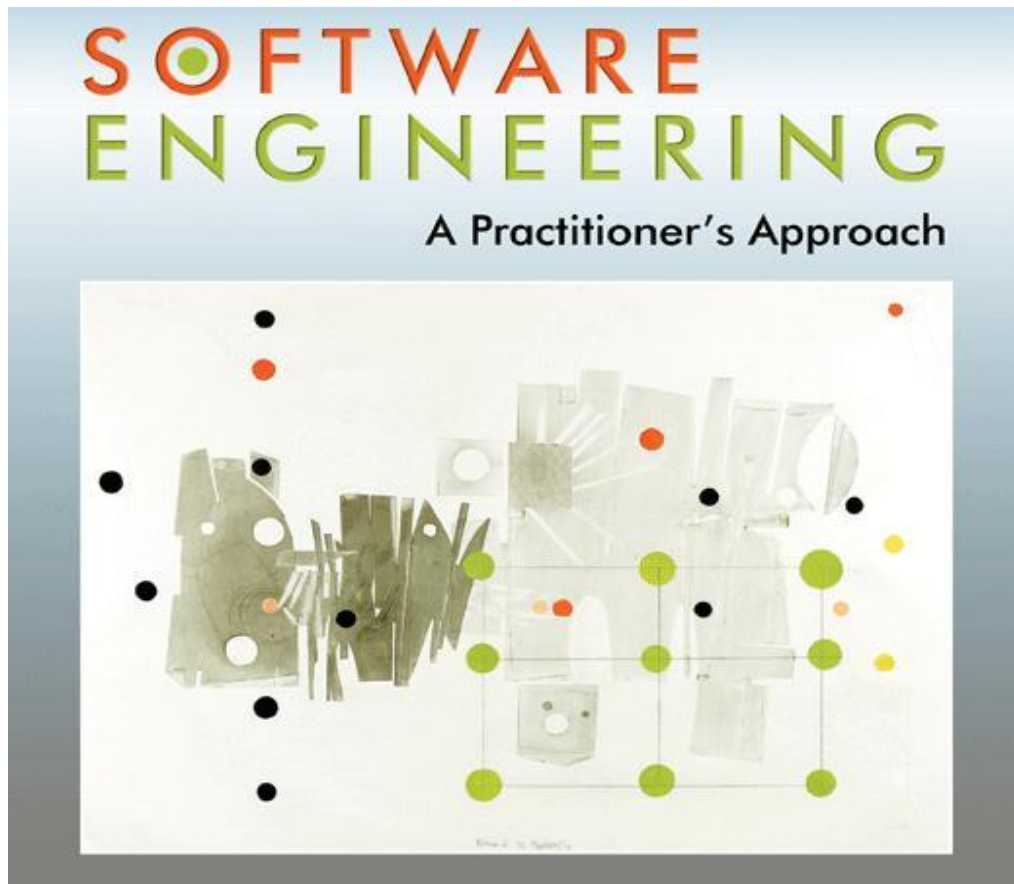


# Программная инженерия



Кафедра  
Информационных технологий и  
СИСТЕМ

---

**Программная инженерия** – это инженерная дисциплина, отражающая все грани разработки программного обеспечения

# Программная инженерия?

---

- **Инженерия** - обеспечивает решение поставленных задач посредством существующих теорий и методов.
- Инженер начинает с **постановки задачи** и поиска инструментов для наилучшего **решения задачи** в рамках существующих **организационных, финансовых и временных ограничений**.
- **Программная инженерия** делает значительный упор на **методы и подходы**, а не на инструменты.

# Программная инженерия (1968-2021)

---

**Программная инженерия** – раздел компьютерных наук (**Computer Science**), изучающий методы и средства построения компьютерных программ как инженерной регламентированной деятельности коллективов разработчиков программных продуктов (ПП):

- определяет объекты (модули, компоненты, аспекты и др.) и операции над ними, объединяющиеся в целостный технологический процесс создания программного продукта;
- обобщает накопленный опыт программирования и отражает закономерности развития технологии программирования. Расширяет содержание понятия “Программирование” путем привлечения понятийного аппарата инженерных дисциплин;
- является *инженерной дисциплиной*, охватывающей все аспекты создания программных продуктов, начиная с формулирования требований, кончая сопровождением и снятием с эксплуатации, а также включает *инженерные методы* управления проектами и оценивание трудозатрат, стоимости и качества изготовления программных продуктов.

Инженерная деятельность по всем аспектам изготовления ПП планируется и декомпозируется на отдельные работы, распределяющиеся по разным категориям исполнителей.

# Computer Science (CS)

- *Компьютерная инженерия* - это методы построения разных вычислительных моделей, компьютеров, механизмов контроля Hardware и Software.
- *Системная инженерия* - это методы построения систем обработки информации, АСУ на основе компьютеров (Computer Systems), принципы их работы и методы управления и выполнения соответствующих классов задач.
- *Программная инженерия* – это теория программирования, инженерия и технология построения компьютерных программных систем.

# Computer science

## Основные дисциплины

### Программная инженерия (Software Engineering)

- ◆ **Теория построения компьютерных систем (КС)**
  - программ для оборудования, оснастки
  - инфраструктура КС
- ◆ **Теория, принципы, концепции программных систем (ПС)**
  - фундаментальные,
  - прикладные
- ◆ **Методы применения, развертывания, конфигурирования КС и ПС**
- ◆ **Програмные методы и технологии**
  - ОС, коммуникационные программы
  - организация вычислений
- ◆ **Прикладные информационные технологии (ИТ)**
  - веб-браузеры, БД, машины поиска
  - теории для нужд организаций
- ◆ **Организационные вопросы информационных систем**
  - теория и принципы организации ИС

### Компьютерная инженерия (Computer Engineering)

- ◆ **Теория построения Hardware Frameworks**
  - компьютеров, микрокомпьютеров, кластеров
  - ПК, суперкомпьютеры (Скит2)
- ◆ **Принципы построения Software для разных типов машин**
  - ОС, управления заданиями
  - СУБД, трансляторов, интерпретаторов, метатрансляторов
- ◆ **Компьютерные архитектуры**
  - микросхемы, микропроцессоры
  - процессоры (Pentium, Intel...)
  - операции последовательные, параллельные, рекурсивные
  - форматы данных, преобразователи
  - интеграция устройств, блоков, карт, кабелей...
  - интеграция процессоров в кластеры)
- ◆ **Фундаментальные теории компьютерной инженерии**
  - теория автоматов
  - машины Тьюринга
  - неймановские машины...

### Системная инженерия (System Engineering)

- ◆ **Построение сложных компьютерных систем**
  - АС, ИС, ИПС, СОД, АСК, ЗДАС...
  - систем реального времени
  - моделирование Computer Application
  - систем програмування
- ◆ **информационные системы**
  - искусственный интеллект
  - управление информацией
  - поисковые системы Интернет
  - БД, БЗ, информационные ресурсы
  - веб-сервисы, веб-семантика
  - електронні бібліотеки
- ◆ **Информационные технологии**
  - интерфейс пользователя
  - технологии взаимодействия
  - процессы обработки, потоки работ
  - накопичення, витяг знань
  - инфраструктура (техника, средства, инструменты, ПЗ, среды)
  - коммуникации и маршрутизации
  - делопроизводство, документооборот
- ◆ **Компьютерные системы**
  - компьютерная графика
  - мультимедиа среды
  - системы защиты информации
  - системы электронного обучения
  - электронная Computer Science
  - прикладные компьютерные системы (экология, медицина, мат. физика вычислительная математика,...)

# ПРОГРАММНАЯ ИНЖЕНЕРИЯ.

## Система дисциплин

### Программирование

- ◆ **Языки и методы**
  - схемы, спецификации
- ◆ **Теоретическое программирование**
  - логическое, функциональное
  - алгоритмическое САА
  - алгебраическое
  - композиционное (номинал.)
  - алгебра, математика
  - теория верификации, доказ.
- ◆ **Прикладное программирование**
  - сборочное,
  - компонентное,
  - агентное, сервисное
  - генерационное
- ◆ **Автоматизация программирования**
  - компиляторы, трансляторы
  - CASE-системы
  - библиотеки, фонды программ
  - верификация, VDM, RSL...
- ◆ **Технология программирования**
  - стадии, этапы, процессы
  - системы (РТК, АПРОП, ПРИЗ, ПРОЕКТ, ППГ)
  - программостроительный ин-т
  - стенды тестирования, контроля, инспекций
  - оценки (размера, надежности)

### Инженерия

- ◆ **Проектирование Application, Domain, Family systems**
  - инженерия требований
  - модели (MDA, DGM, Mxap, Mпро, PIM)
  - use case, UML
  - КПИ (reuse, артефакты)
  - ЖЦ модельного подхода
- ◆ **Тестирование**
  - трассирование требований
  - тестирование как процесс ЖЦ
  - поиск ошибок, отказов,
  - интенсивность отказов, надежность
  - тестирование с учетом рисков отказов
- ◆ **Качество**
  - ядро знаний в области качества
  - парадигма качества
  - модели качества (трехмерная, стандарт.)
  - система управления качеством
- ◆ **Измерение и оценивание**
  - парадигма оценивания
  - объектов (целевых, объектов ЖЦ)
  - качества, надежности, стоимости
  - процессов ЖЦ, базового процесса
  - аттестация программного продукта
- ◆ **Реинженерия, реверсная инженерия**
- ◆ **Стандарты по программной инженерии**
  - ЖЦ (ISO/IEC 12207)
  - оцінки (ISO/IEC 14598 (1-4))
  - вимірювання (ДСТУ ISO/IEC 15939)
  - якості (ISO 9000 (1-4), ISO/IEC 9126)
  - SWEBOK, PMBOK

### Экономика

- ◆ **Аналитические расчеты**
  - затрат, размера
  - модели (Cosomo, FPA)
- ◆ **Оценка трудоемкости**
- ◆ **Оценка стоимости работ и ПС**

### Управление

- ◆ **Управление программными проектами**
  - методы (CRM, PERT, Gant)
  - планирование, контроль, оценки
- ◆ **Управление рисками**
  - таксономия рисков, контроль
- ◆ **Управление конфигурацией**
  - контроль версий и изменений в ПС

### Индустрия

- ◆ **Технологическая подготовка разработки (ТПР)**
  - технол. линии (ТЛ) и процессы (ТП)
  - инфраструктура ТПР
  - фонды алгоритмов и программ
  - служба качества и контроля
- ◆ **Линии производства продуктов**
  - ТЛ, инструменты разработки, сборки, оценивания
  - мониторинг программного продукта
  - библиотеки (Matlab, Greed, IP)
  - Интернет ресурсы (Web,ECS, Grid)

# Направления работ в области программной инженерии

1. SWEВОК, РМВОК (2001, 2004).
2. Объектно-ориентированное (UML), компонентное программирование (KM), сервисное, генерационное.
3. Фундаментальные типы данных (FDT) ЯП – Ч.Хоар, Вирт (1974), В.Н.Агафонов (1981), Типы данных общего назначения (GDT) – Стандарт ISO/IEC 11404-1996, 2007 ...
4. Стандарты ISO/IEC 12207- ЖЦ, ДСТУ 9126 - качество ...
5. Инструментальные среды разработки ПС (COM, CORBA, MS.VSTS, JAVA, MSF, Grid, Oberon, Babel, Rational Rose...).
6. Технологические линии, Product lines.
7. Дисциплины индустрии ПП (наука, инженерия, экономика, управление и др.)



# Классификация дисциплин SE



# ПРОГРАММНАЯ ИНЖЕНЕРИЯ

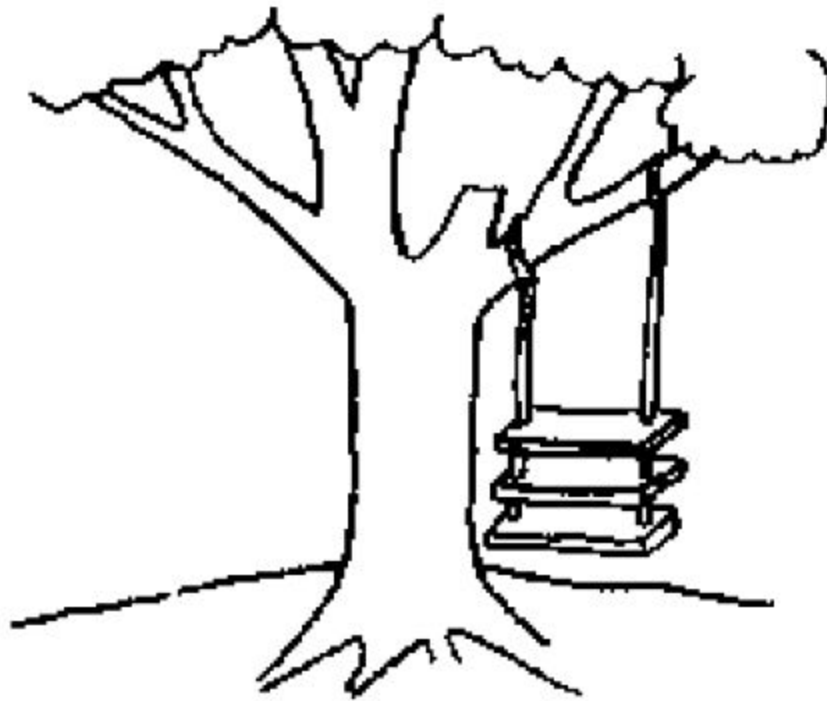


**Заказчик**

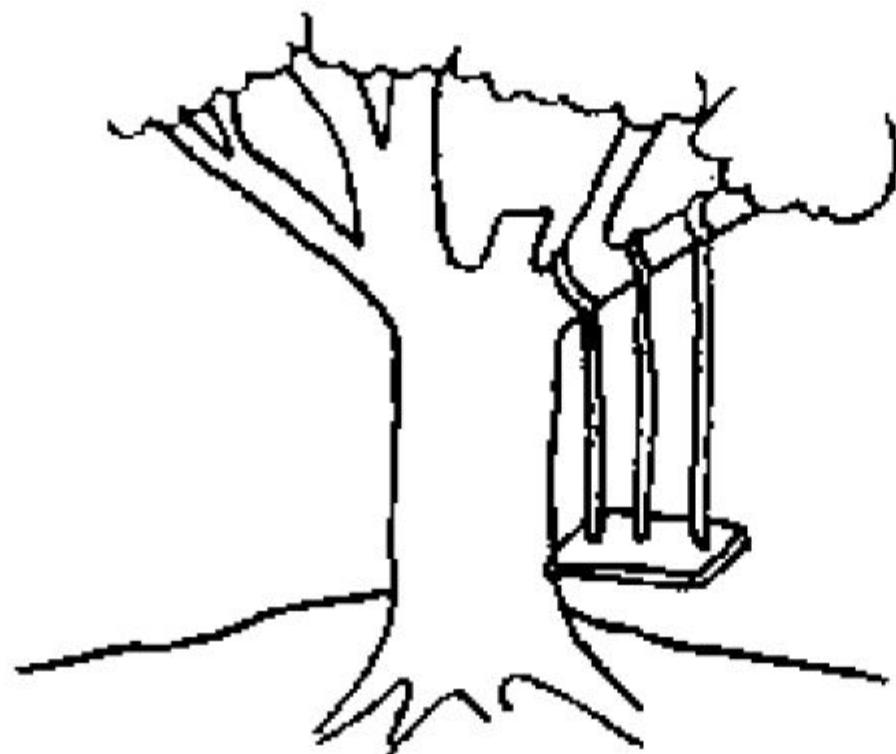
Задача,  
требования,  
ограничения

**Программист**

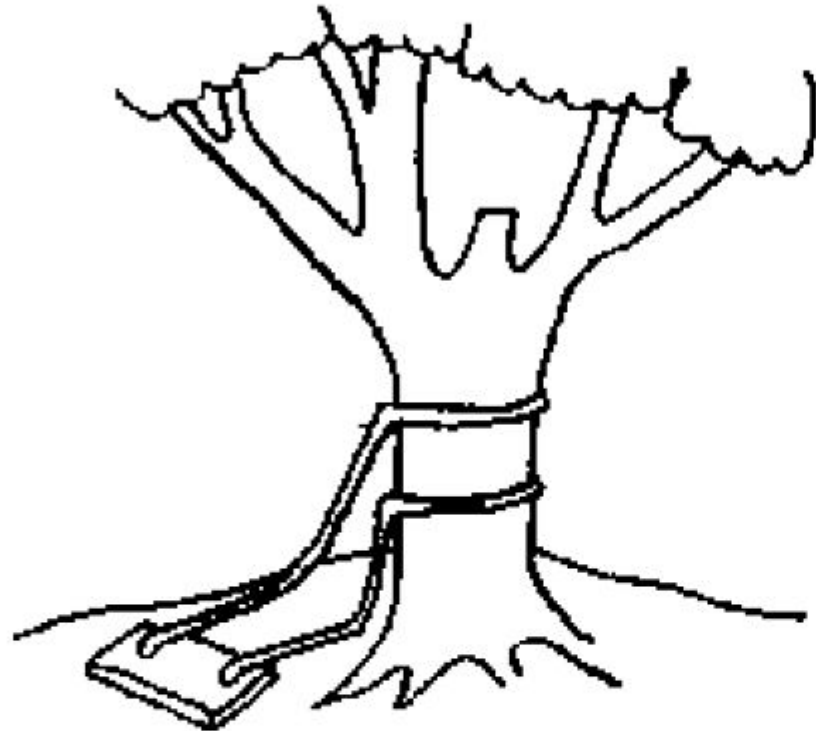
## «КАЧЕЛИ» - КАК ПРОЕКТИРУЮТСЯ ПРОГРАММЫ ( 1975 ! )



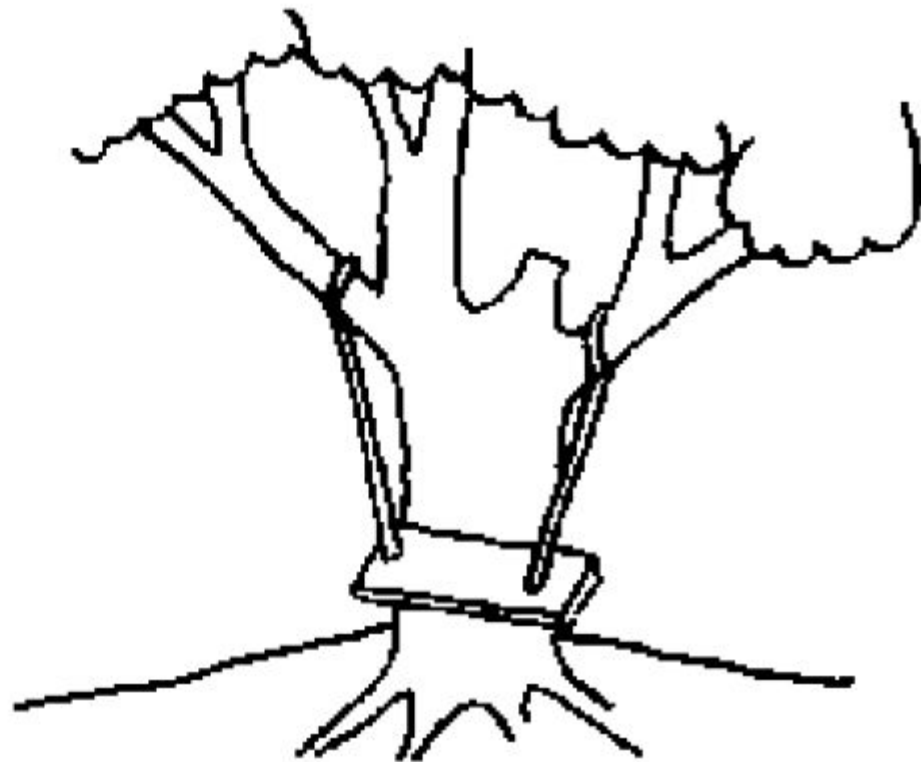
**Как было предложено  
организатором разработки**



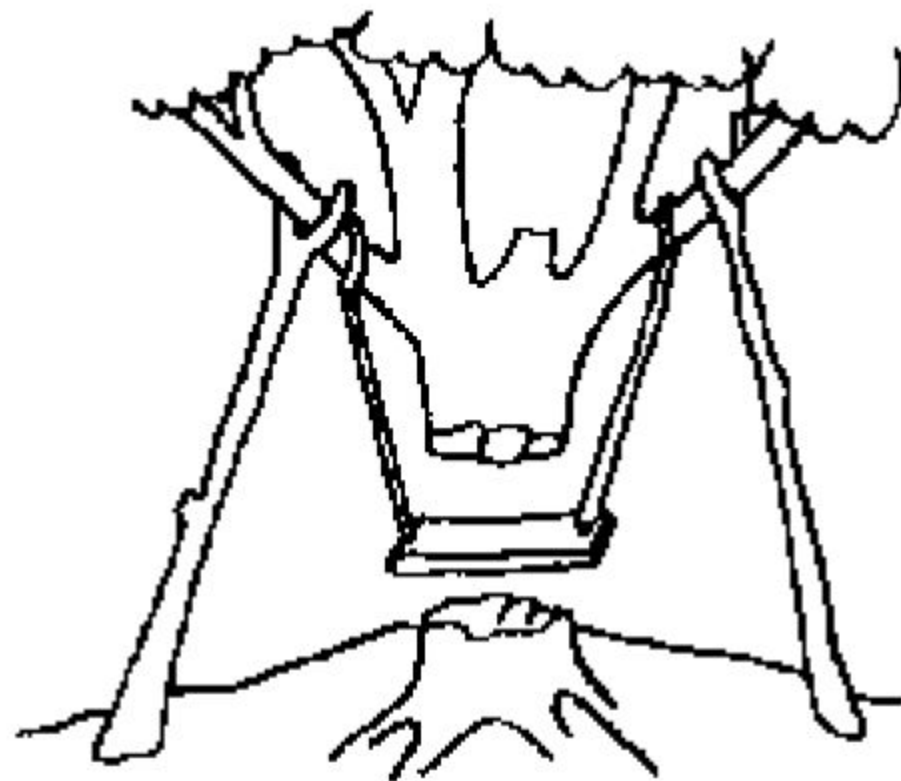
**Как было описано  
в техническом задании**



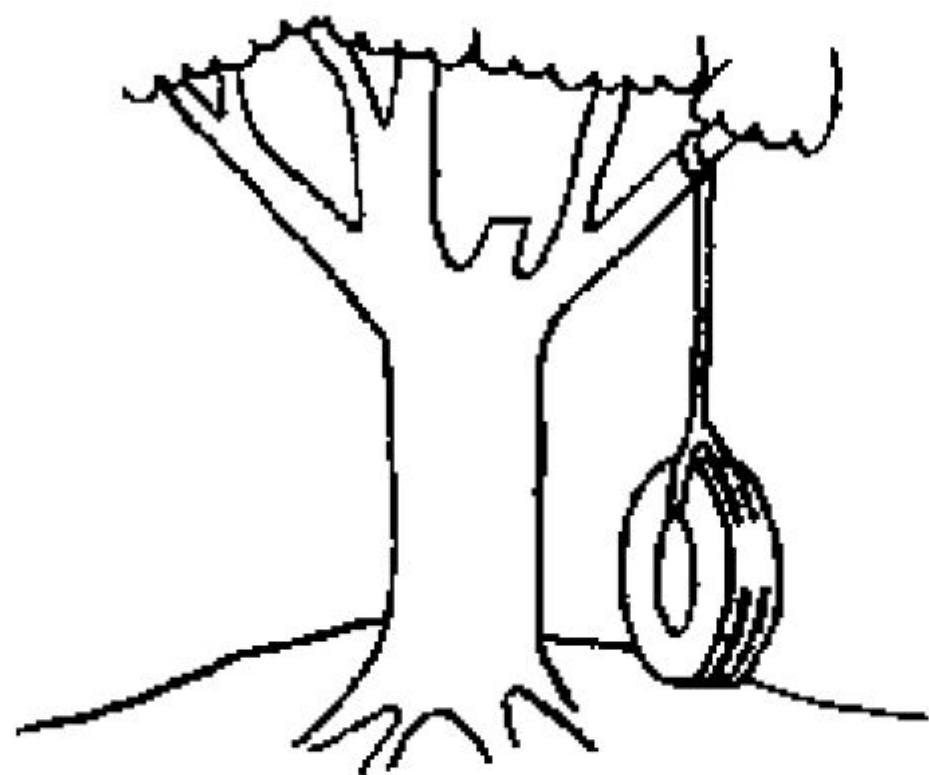
**Как было спроектировано  
ведущим системным специалистом**



**Как было реализовано  
программистами**



**Как было внедрено**



**Чего хотел пользователь**



---

# **Тема 1. Программная инженерия: назначение, основные принципы и понятия**

---

# Предпосылки и история

---

- ❑ Повторное использование кода
  - Модульное программирование
- ❑ Рост сложности программ
  - Структурное программирование
- ❑ Модификация программ
  - Объектно-ориентированное программирование

# Повторное использование кода

---

- ❑ Проблема
  - Дублирование фрагментов кода
- ❑ Модульное программирование
  - Выделение фрагментов в модули
  - Повторное использование модулей
  - Создание библиотек модулей

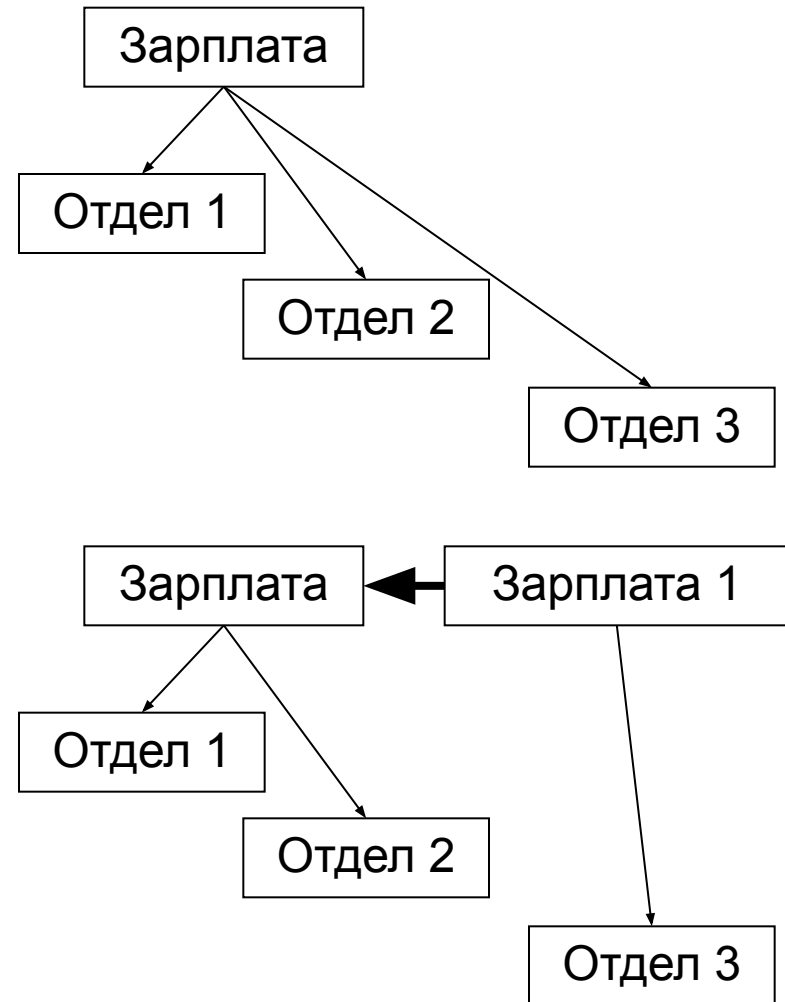
# Рост сложности программ

---

- Проблема
  - Сложные программные комплексы
    - Объем кода, к-во связей, к-во разработчиков, к-во пользователей
  - Жизненный цикл: стадии внедрения и сопровождения
  
- Структурное программирование
  - «Правильное» проектирование и кодирование
  - Основные принципы:
    - Нисходящее проектирование
    - Применение специальных языков проектирования
    - Дисциплина проектирования и разработки:
      - планирование и документирование проекта
      - поддержка соответствие кода проектной документации
    - Структурное кодирование (линейный блок, If-then-else, цикл)

# Модификация программ

- ❑ Проблема
  - изменения в проекте и программе без изменения ранее написанного кода
- ❑ Объектно-ориентированное программирование
  - Класс – модуль со свойствами, поведением, обязанностями
  - Парадигмы ООП:
    - Инкапсуляция и сокрытие деталей
    - Наследование
    - Полиморфизм



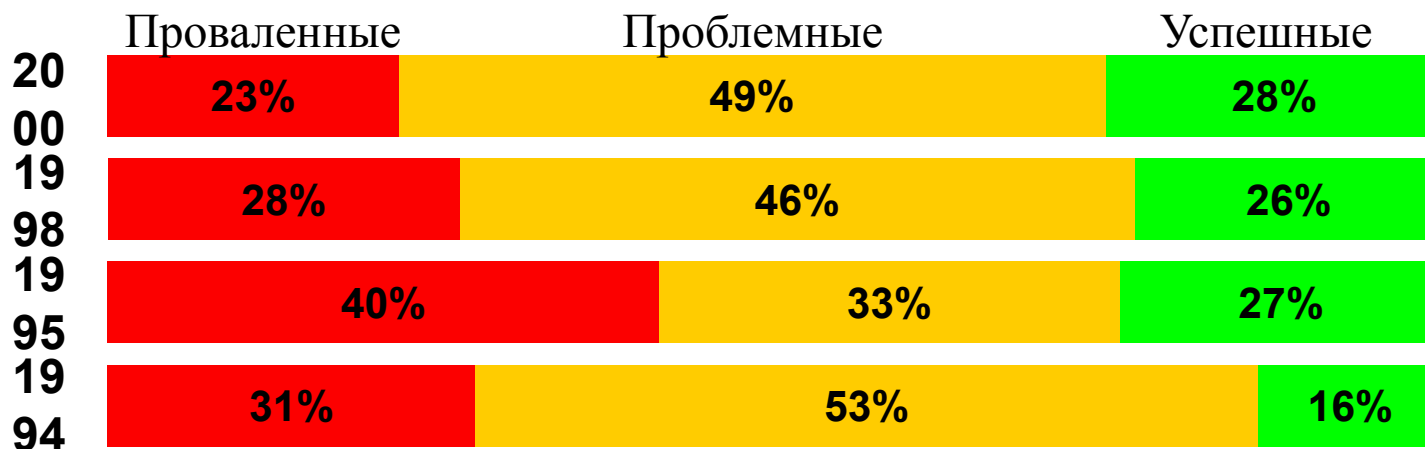
# Некоторые итоги

---

- ❑ Главная цель программной инженерии - сокращение стоимости ПО
- ❑ Сформировались основные принципы и методы проектирования ПО:
  - Жизненный цикл ПО
  - Модульное программирование
  - Структурное проектирование и программирование
  - Объектно-ориентированное проектирование и программирование

# Кризис программирования

- ❑ Кризис программирования принимает хронические формы:
  - США тратит более \$200 млрд. на более чем 170 тыс. проектов
  - потери от недополученного эффекта измеряются триллионами.
- ❑ Успешные проекты не часты (30000 проектов)



Источник: The Standish Group International, Inc., Extreme Chaos, 2000  
[http://www1.standishgroup.com/sample\\_research/PDFpages/extreme\\_chaos.pdf](http://www1.standishgroup.com/sample_research/PDFpages/extreme_chaos.pdf)

# Начнем с определений

---

- Программная инженерия – это
  - установление и использование обоснованных инженерных принципов (методов) для экономного получения ПО, которое надежно и работает на реальных машинах. [Bauer 1972].
  - та форма инженерии, которая применяет принципы информатики (computer science) и математики для рентабельного решения проблем ПО. [CMU/SEI-90-TR-003]
  - применение систематического, дисциплинированного, измеряемого подхода к разработке, использованию и сопровождению ПО [IEEE 1990].
  - дисциплина, целью которой является создание качественного ПО, которое завершается вовремя, не превышает выделенных бюджетных средств и удовлетворяет выдвигаемым требованиям [Schach, 99]



# Разберемся в вопросах

---

- ❑ Что такое программное обеспечение (software)?
- ❑ Что такое программная инженерия?
- ❑ В чем разница между программной инженерией (software engineering) и информатикой (computer science)?
- ❑ В чем отличие программной инженерии от других инженерий?
- ❑ Из чего складывается стоимость ПО?

# Программное обеспечение?

---

- ❑ Компьютерные программы и связанная с ними документация и данные (ISO/IEC 12207)
- ❑ Программные продукты могут разрабатываться для конкретного заказчика или для обобщенного рынка
- ❑ Программные продукты могут быть
  - Коробочными (generic products, shrink-wrapped software), т.е. разработанными для продажи многим различным заказчикам
  - Заказными (bespoke, custom), т.е. разработанными для одного покупателя по его спецификациям

# Программная инженерия?

---

- ❑ Инженерная дисциплина, которая связана со всеми аспектами производства ПО
  - от начальных стадий создания спецификации до поддержки системы после сдачи в эксплуатацию
- ❑ Инженерная дисциплина
  - Ориентация на практический результат
  - Применение теорий, методов и способов для достижения результата
  - Лучшие практики (best practices)
  - При ограниченном ресурсе времени, бюджета, оборудования, людей
- ❑ Все аспекты производства ПО
  - Управление программными проектами
  - Разработка средств, методов и теорий

# В чем отличия от информатики?

---

- ❑ Информатика (computer science) - теория и методы вычислительных и программных систем
- ❑ Программная инженерия - практические проблемы создания ПО
- ❑ Информатика – теоретический фундамент программной инженерии
  - Не всегда достаточный
  - Не единственный (финансы, управление проектом, работа с заказчиком, ...)

# В чем отличие от других инженерий?

- ❑ Вопросы:
  - Почему так велика доля провальных проектов?
  - Можно ли применять опыт других инженерий?
- ❑ Фазы жизненного цикла любого продукта:
  - Проектирование, создание образца, испытание, производство, эксплуатация
- ❑ Программа – не материальный объект:
  - Фазы производства и изготовления образца отсутствуют
  - Стоимость программы – это стоимость проектирования
  - У коробочных продуктов «размазывается» по копиям
  - У заказных продуктов остается высокой.

# В чем еще отличие от др. инженерий?

---

- ❑ Программа – искусственный объект
  - Нет объективных законов контроля проекта
  - Тестирование – единственный способ проверки
- ❑ Программная инженерия – молодая дисциплина
- ❑ Подробнее
  - Кони Бюрер «От ремесла к науке: поиск основных принципов разработки ПО»  
<http://interface.ru/fset.asp?Url=/rational/science.htm>

# Из чего складывается стоимость ПО?

---

- ❑ Зависит от типа ПО, методологии разработки и ... метода оценки
- ❑ Типовое распределение:
  - 15% - спецификация
  - 25% - проектирование
  - 20% - разработка
  - 40% - интеграция и тестирование
- ❑ Коробочное ПО
  - Рост доли тестирования за счет спецификации
- ❑ Заказное ПО
  - Рост доли тестирования за счет проектирования и разработки

# Еще вопросы

---

- ❑ Что такое программный процесс?
- ❑ Что такое модель программного процесса?
- ❑ Что такое методы программной инженерии?
- ❑ Что такое CASE (Computer-Aided Software Engineering)?
- ❑ Какими свойствами обладает хорошая программа?
- ❑ Какие основные трудности стоят перед программной инженерией?



# Программный процесс?

---

- ❑ Жизненный цикл – непрерывный процесс с момента принятия решения о создании ПО до снятия его с эксплуатации.
- ❑ Процесс – совокупность действий и задач, имеющих целью достижение значимого результата.
- ❑ Основные процессы (этапы или фазы) ЖЦ:
  - Спецификация требований
  - Разработка проекта программы
  - Кодирование
  - Тестирование
  - Документирование

# Программный процесс?

---

- Дополнительные (нефункциональные) процессы:
  - создание инфраструктуры, управление конфигурацией, управление качеством, обучение, разрешение противоречий, ...
  
- Установление процесса:
  - Описание процесса
  - Обучение процессу
  - Введение метрик
  - Контроль выполнения
  - Усовершенствование

# Модель программного процесса?

---

- ❑ Модель программного процесса — это упрощенное описание программного процесса, представленное с некоторой точки зрения.
- ❑ Модели жизненного цикла:
  - Водопадная (каскадная) модель
  - Спиральная (циклическая) модель
  - Компонентная модель
  - Формальная модель
  - Комбинированные модели
- ❑ Модели организации работ:
  - Модель потока работ (workflow model)
  - Модель потоков данных (data flow model)
  - Ролевая модель

# Методы программной инженерии?

---

- ❑ Метод программной инженерии — это структурный подход к созданию ПО:
  - как высококачественного продукта
  - экономически эффективным способом.
  
- ❑ Наиболее известные методы:
  - Структурного анализа и проектирования Том ДеМарко (1978),
  - Сущность-связь Чен (1976)
  - Объектно-ориентированного анализа и проектирования Буч (1994), Рамбо (1991).

# Методы программной инженерии?

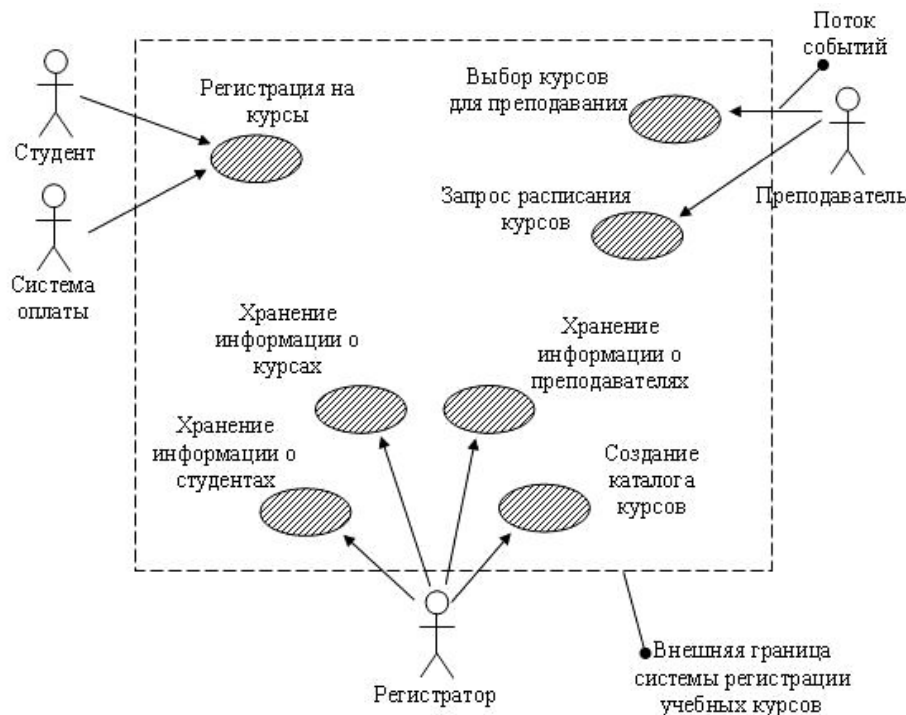
---

- ❑ Цель - создание и поэтапное преобразование моделей ПО
- ❑ Методы должны включать в себя следующие компоненты:
  - Описание моделей системы и нотация
  - Правила и ограничения
  - Рекомендации
  - Руководство по применению метода
- ❑ Нет идеальных методов, нет абсолютных методов

# Модель прецедентов (требований)

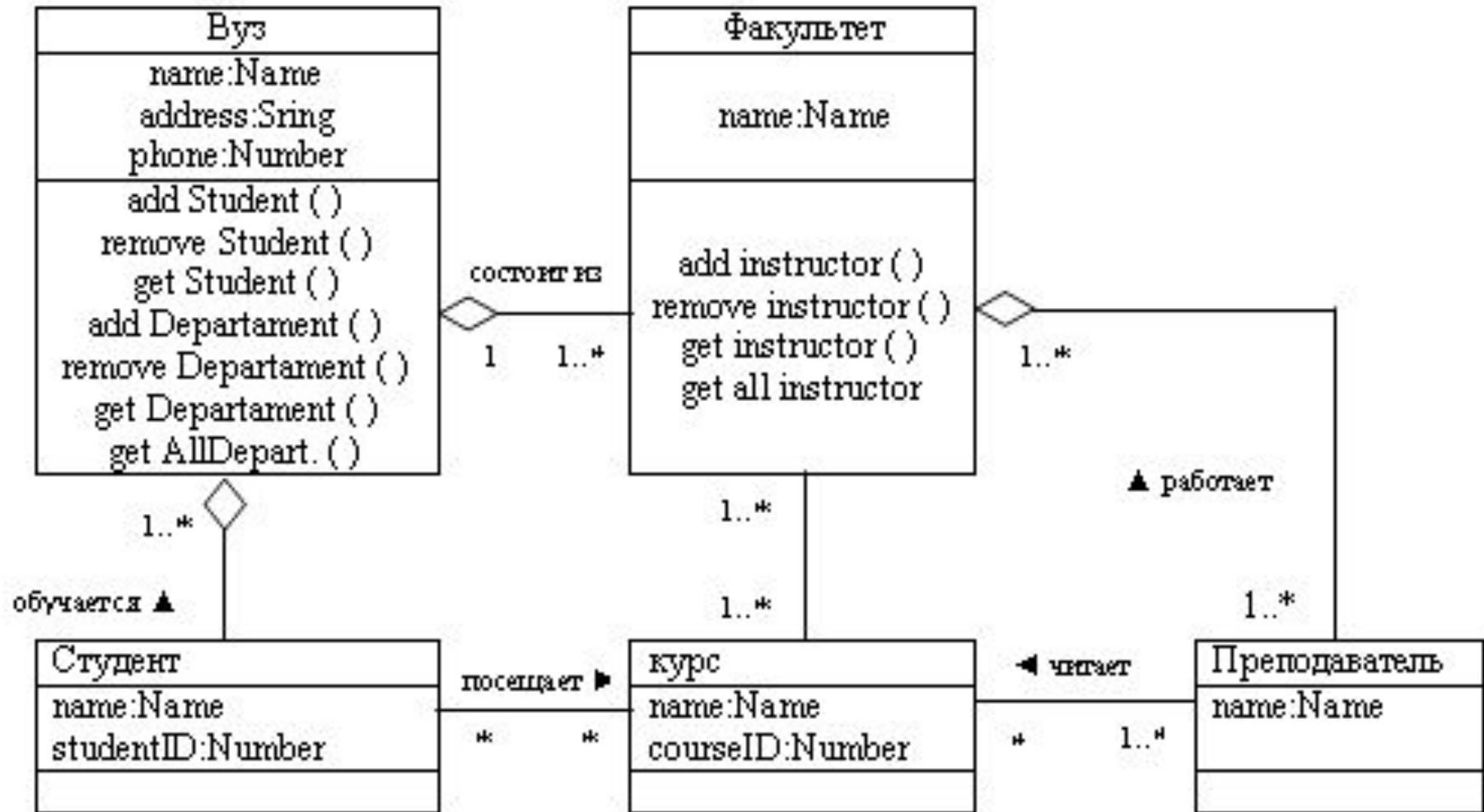
## Описание прецедента «Выбор курса для преподавателя»

- Прецедент начинается выполняться, когда преподаватель подключится к системе. Преподаватель вводит нужный семестр. Система предлагает выбрать требуемую операцию: добавить (Add), удалить (Delete), просмотреть (Review), напечатать (Print) или выйти (Quit).

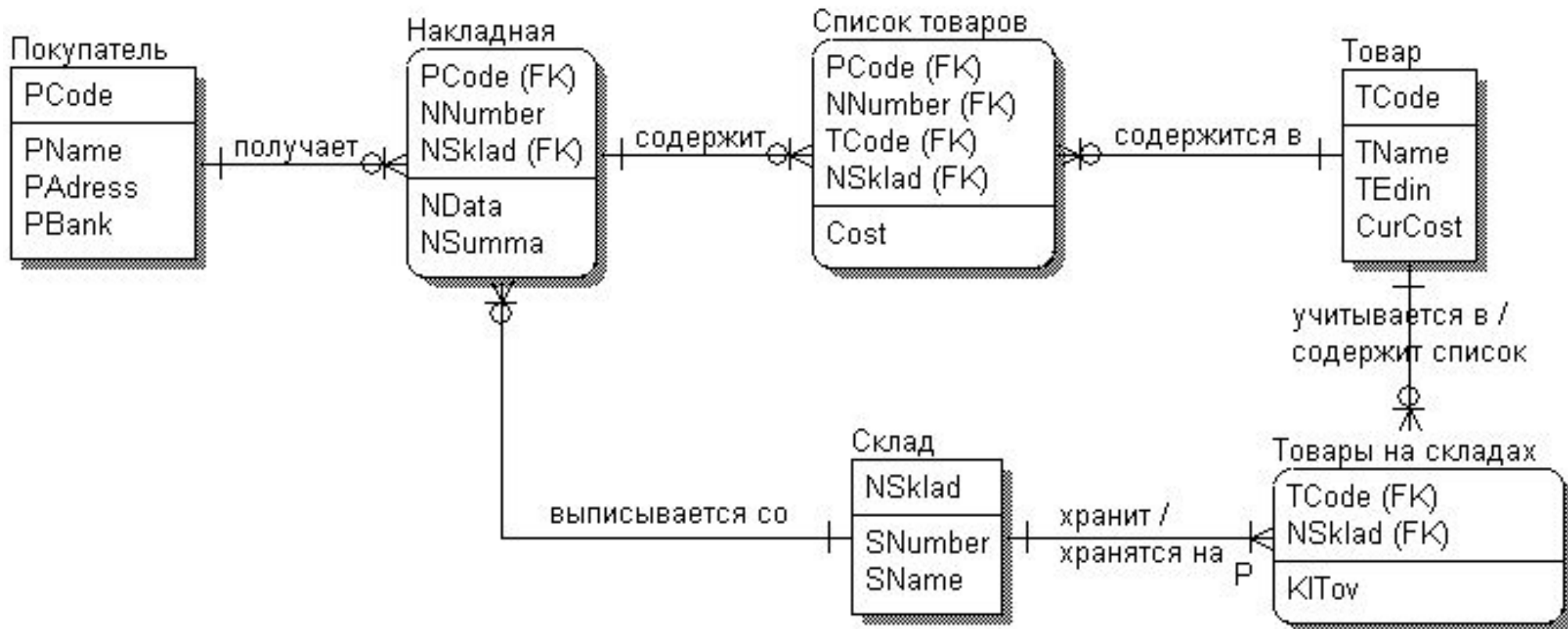


- Если выбрана операция добавить (Add), S-1: выполняется поток добавить учебный курс (Add a Course Offering).
- Если выбрана операция удалить (Delete), S-2: выполняется поток удалить учебный курс (Delete a Course Offering).
- Если выбрана операция просмотреть (Review), S-3: выполняется поток просмотреть расписание (Review Schedule).
- Если выбрана операция ....

# Модель (диаграмма) классов



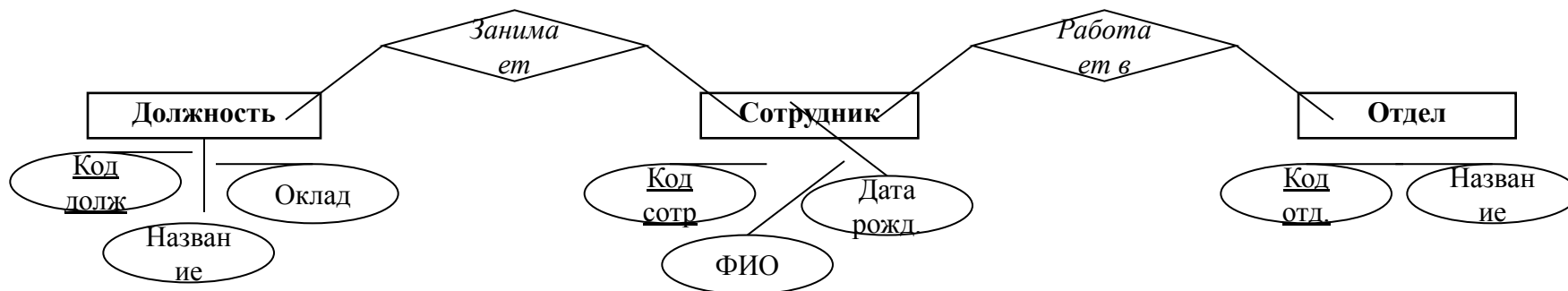
# Модель сущность-связь



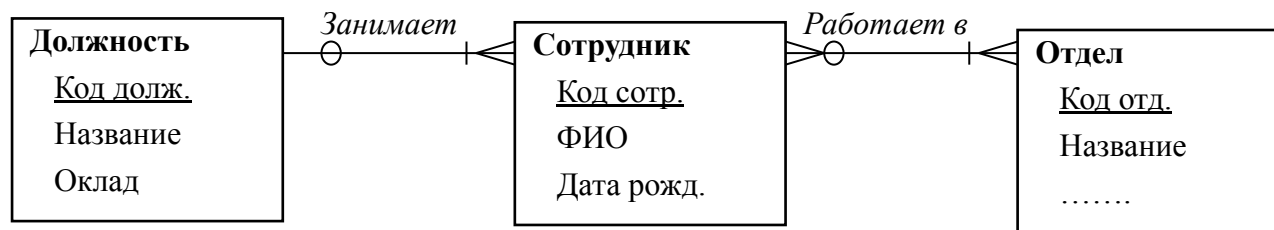


# Нотации модели

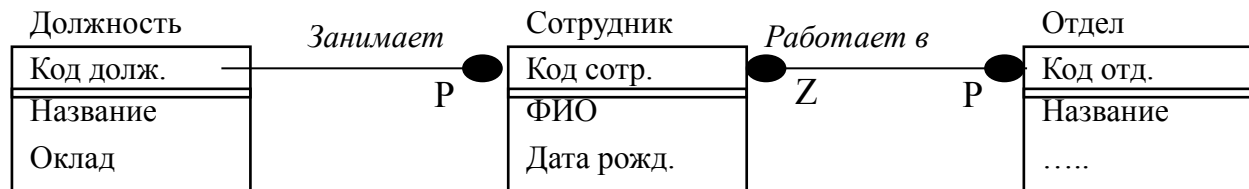
Нотация  
Чена



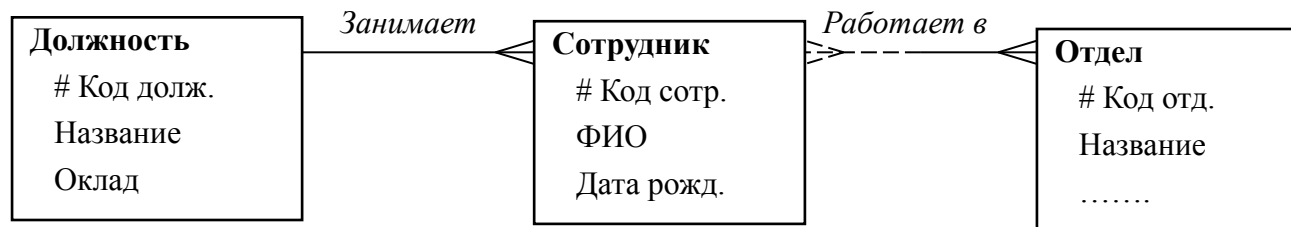
Нотация  
Мартина



Нотация  
IDEF1X



Нотация  
Баркера



# Определение CASE



- ❑ Использование компьютеров для поддержки процесса создания программ. Может включать инструментальные программные средства для проектирования ПО, формирования требований, кодирования, тестирования, документирования и других действий по созданию ПО
- ❑ Эволюция CASE
  - Ассемблеры, компиляторы, интерпретаторы
  - Трассировщики, символические отладчики
  - Текстовые редакторы, системы анализа исходных текстов
  - Средства для анализа требований и проектирования
  - Среды визуального программирования
  - Средства генерации исходных кодов
  - Средства поддержки полного жизненного цикла ПО

# Классификация CASE

---

- Классификация CASE средств:
  - По уровню применения:
    - Upper CASE - средства анализа требований
    - Middle CASE - средства проектирования
    - Low CASE - средства разработки приложений
  - Специализированные
    - Средства проектирования БД
    - Средства реинжиниринга
  - Вспомогательные
    - Планирования и управления проектом
    - Конфигурационного управления
    - Тестирования
- Интегрированные CASE
- Главное правило: сначала метод – потом CASE

# Свойства хорошей программы

---

- ❑ Удовлетворять функциональным требованиям
- ❑ Нефункциональные требования:
  - Сопровождаемость (maintainability)
    - Возможность дальнейшего развития.
  - Надежность (dependability)
    - Отказоустойчивость, безопасность, защищенность
  - Эффективность (efficiency)
    - Память, процессорное время, каналы связи.
  - Удобство использования (usability)
    - Понятно пользователю

# Основные трудности

---

- ❑ Главная проблема: универсальный метод и процесс
- ❑ Основные трудности:
  - Наследование ранее созданного ПО (legacy systems).
    - Сопровождение – поддержка и развитие старого ПО.
  - Разнородность программных систем.
    - Распределенные сети, разнородное оборудование, разные среды, различные ОС
  - Сокращение времени на разработку.
    - Сократить время разработки ПО без снижения его качества.
- ❑ Трудности часто оказываются связанными между собой

# Профессиональные и этические требования

---

- ❑ Развитие IT индустрии оказывает все большее воздействие на общество
  - Internet, телекоммуникации, IP телефония, компьютерные игры ...
- ❑ IT специалисты работают в правовом и социальном окружении, под действием международных, национальных и местных законодательств.

# Профессиональные и этические требования

---

- Более тонкие профессиональные обязательства:
  - Конфиденциальность
    - Неразглашение сведений о своих работодателях или заказчиков независимо от того, подписывалось ли ими соответствующее соглашение.
  - Компетентность
    - не должен завышать свой уровень компетентности и браться за работу, не соответствующую этому уровню
  - Защита интеллектуальной собственности
    - соблюдать законодательство при использовании чужой интеллектуальной собственности
    - защищать интеллектуальную собственность работодателя и клиента
  - Злоупотребление компьютером
    - от игр в компьютерные игрушки на рабочем месте до распространения вирусов и т.п.

# Кодекс этики IEEE-CS/ACM

---

- ❑ ACM, IEEE и British Computer Society
  - IEEE-CS/ACM Software Engineering Code of Ethics and Professional Practices – Кодекс этики и профессиональной практики программной инженерии..
- ❑ Члены этих организация принимают на себя этот кодекс
- ❑ Кодекс содержит восемь Принципов, связанных с поведением и решениями, принимаемыми профессиональными программистами
- ❑ Кодекс распространяется также на студентов и «подмастерьев», изучающих данную профессию
- ❑ Кодекс имеет краткую и полную версии



# Кодекс этики - Преамбула

---

- Краткая версия кодекса
  - суммирует стремления кодекса на высоком уровне абстракции.
  - полная версия показывает как эти стремления отражаются на деятельности профессиональных программистов.
  - без высших принципов детали кодекса станут казуистическими и нудными;
  - без деталей стремления останутся возвышенными, но пустыми;
  - вместе же они образуют целостный кодекс.
  
- Программные инженеры должны добиваться, чтобы анализ, спецификация, проектирование, разработка, тестирование и сопровождение программного обеспечения стали полезной и уважаемой профессией. В соответствии с их приверженностью к процветанию, безопасности и благополучию общества, программные инженеры будут руководствоваться следующими Восемью Принципами

# Кодекс этики: 8 принципов

---

## 1. ОБЩЕСТВО

- Программные инженеры будут действовать соответственно общественным интересам.

## 2. КЛИЕНТ И РАБОТОДАТЕЛЬ

- Программные инженеры будут действовать в интересах клиентов и работодателя, соответственно общественным интересам.

## 3. ПРОДУКТ

- Программные инженеры будут добиваться, чтобы произведенные ими продукты и их модификации соответствовал высочайшим профессиональным стандартам.

# Кодекс этики: 8 принципов

---

## 4. СУЖДЕНИЕ

- Программные инженеры будут добиваться честности и независимости в своих профессиональных суждениях

## 5. МЕНЕДЖМЕНТ

- Менеджеры и лидеры программных инженеров будут руководствоваться этическим подходом к руководству разработкой и сопровождением ПО, а также будут продвигать и развивать этот подход

## 6. ПРОФЕССИЯ

- Программные инженеры будут улучшать целостность и репутацию своей профессии соответственно с интересами общества

# Кодекс этики: 8 принципов

---

## 4. КОЛЛЕГИ

- Программные инженеры будут честными по отношению к своим коллегам и будут всячески их поддерживать

## 8. ЛИЧНОСТЬ

- Программные инженеры в течение всей своей жизни будут учиться практике своей профессии и будут продвигать этический подход к практике своей профессии

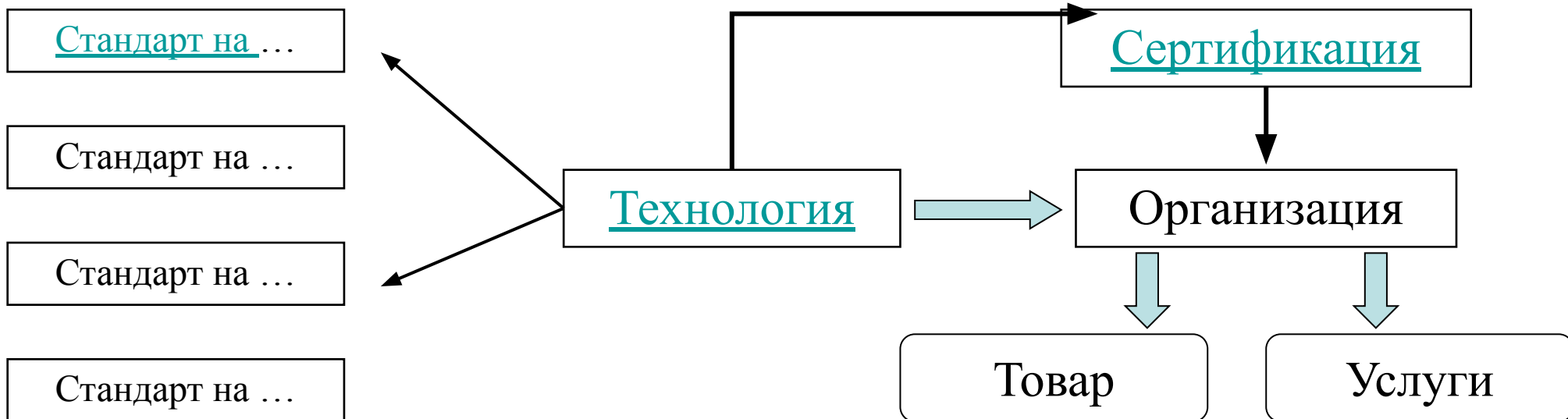
Полная версия кодекса: IEEE-CS/ACM Software Engineering Ethics and Professional Practices.  
<http://www.computer.org/tab/seprof/code.htm#Public>

# Стандартизация и стандарты

---

- ❑ Что такое стандарты и сертификация на соответствие стандартам?
- ❑ Какие бывают стандарты?
- ❑ Кто разрабатывает стандарты программной инженерии?
- ❑ Основные стандарты программной инженерии

# Стандарты и сертификация





- Знание (логия) мастерства (техно):
  - совокупность приёмов и способов производства;
  - научная дисциплина
  - сами операции производства
  - описание производственных процессов,
  - инструкции по их выполнению,
  - технологические правила, требования, карты, графики и др.

# Что такое стандарт?



- ❑ От англ. standard - норма, образец, мерило:
  - нормативный документ, устанавливающий комплекс норм и правил;
  - типовой образец, эталон, модель,
- ❑ Стандарт может быть разработан на
  - материально-технические предметы
  - нормы, правила, требования
- ❑ Стандартизация распространяется на все сферы человеческой деятельности:
  - науку, технику, промышленное и с.-х. производство, строительство, здравоохранение, транспорт и т.д.



# Что такое сертификация?



- ❑ Certification - "сделано верно“:
- ❑ Заявление о соответствии поставщика (изготовителя) (supplier's declaration)
  - письменная гарантия соответствия продукции заданным требованиям
- ❑ Сертификация соответствия
  - процедура, посредством которой третья сторона дает письменную гарантию, что продукция, процесс, услуга соответствуют заданным требованиям.

# Какие бывают стандарты?

---

- Типы стандартов
  - Корпоративные
  - Отраслевые
  - Государственные
  - Международные
  
- Виды стандартов
  - Рекомендательные
  - Обязательные

# Кто разрабатывает стандарты SE?

Сокр.	Name	Название
<a href="#"><u>ISO</u></a>	International Organization for Standardization	Международная организация по стандартизации
<a href="#"><u>ACM</u></a>	Association for Computing Machinery	Ассоциация вычислительной техники
<a href="#"><u>SEI</u></a>	Software Engineering Institute	Институт программного инжиниринга
<a href="#"><u>PMI</u></a> ®	Project Management Institute	Институт управления проектами
<a href="#"><u>IEEE</u></a>	Institute of Electrical and Electronics Engineers	Институт инженеров по электротехнике и электронике
ANSI	American National Standards Institute	Американский национальный институт стандартов



- ❑ Международная организация по стандартизации (ISO)
- ❑ Неправительственная организация
  - Создана в 1947 году.
  - Консультативный статус ООН
  - Федерация национальных организаций (146 комитетов-членов ISO)
  - Разработка стандартов в технических комитетах (188 комитетов и подкомитетов)



- ❑ Стандарты ISO - рекомендательные
  - Нет контроля за выполнением стандартов, нет сертификации на соответствие
  - Использование логотипа ISO незаконно
- ❑ Принятие проекта
  - Требуется 75% голосов комитетов-членов
- ❑ Подробнее:
  - <http://www.russianregister.com.ua/page18.html>



- ❑ Ассоциация по вычислительной технике
- ❑ Основана в 1947г.
- ❑ Крупнейшая всемирная научная и образовательная организация
  - Более 75000 профессионалов компьютерной науки
  - До 100 международных конференций в год
  - Несколько десятков научных журналов
  - Большое количество авторитетных наград и премий
- ❑ Разработка учебных программ и стандартов
- ❑ Подробнее: <http://www.acm.org/>



- ❑ Институт Программной Инженерии в университете Карнеги-Меллона
  - Федеральное финансирование
  - Заказчик - министерство обороны США
- ❑ Основные задачи:
  - Методики оценки уровня зрелости организаций
  - Разработки в области IT и SE
  - Методики разработки высококачественного ПО
- ❑ Доступ к самым передовым техническим инновациям
- ❑ Подробнее: [www.sei.cmu.edu](http://www.sei.cmu.edu)



- ❑ Международный Институт Проектного Менеджмента
- ❑ Основан в 1969 г. (США, Филадельфия)
- ❑ Международная общественная организация
  - от 100000 до 135000 членов в 125 странах
  - продвижение, пропаганда, развитие проектного менеджмента
  - разработка стандартов проектного менеджмента
  - повышение квалификации и сертификация специалистов
- ❑ Исследования в области проектного менеджмента
  - конференции, гранты, исследовательская база данных и т.д.
  - издание журналов, в онлайн-магазине более 1000 наименований.
- ❑ Подробнее: <http://www.pmi.org> и <http://www.pmi.ru>





- ❑ Институт инженеров по электротехнике и электронике
  - 400000 специалистов из более чем 150 стран
  - Состоит из ряда профессиональных сообществ
- ❑ Самое крупное из которых - IEEE Computer Society
  - более 100000 человек
  - ежегодно около 150 конференций и симпозиумов
  - более 20 периодических изданий
  - деятельность по стандартизации (200 рабочих групп)
- ❑ Подробнее: <http://www.ieee.org> и <http://www.computer.org.ru/>

# Основные стандарты SE

---

- ❑ [ISO/IEC 12207](#) - Information Technology - Software Life Cycle Processes
- ❑ [SEI CMM](#) - Capability Maturity Model (for Software)
- ❑ [ISO/IEC 15504](#) - Software Process Assessment
- ❑ [PMBOK](#) - Project Management Body of Knowledge
- ❑ [SWEBOK](#) - Software Engineering Body of Knowledge
- ❑ [ACM/IEEE CC2001](#) - Computing Curricula 2001



- ❑ ГОСТ Р ИСО/МЭК 12270. 2000
  - Процессы жизненного цикла программных средств
- ❑ Программный продукт (или ПО):
  - набор компьютерных программ, процедур
  - и связанной с ними документации и данных
- ❑ Жизненный цикл ПО – это непрерывный процесс, который
  - начинается с момента принятия решения о необходимости его создания
  - заканчивается в момент его полного изъятия из эксплуатации
- ❑ Стандарт определяет:
  - организацию и
  - структуру ЖЦ ПО



- ❑ Capability Maturity Model (for Software) - модель зрелости процессов разработки ПО
- ❑ Как выбирать организацию, которой можно доверить выполнение крупного IT проекта?
- ❑ Пять уровней зрелости процесса
  1. Начальный (Initial)
  2. Повторяемый (Repeatable)
  3. Определенный (Defined)
  4. Управляемый (Managed)
  5. Оптимизируемый (Optimized)



- ❑ ISO/IEC 15504 TR Software Process Assessment
  - SPICE: Software Process Improvement and Capability dEtermination
  - Оценка и аттестация зрелости процессов создания и сопровождения ПО.
- ❑ Опыт 9 стандартов (ISO 12207, CMM, ...)
  - Расширенное количество процессов (ISO 12207)
  - 6 уровней зрелости (CMM)
- ❑ Аттестация ❑ оценка зрелости, усовершенствовов.
- ❑ Регламенты:
  - Аттестации, усовершенствовования, оценки
  - Компетентности аттестаторов



- ❑ Project Management Body of Knowledge - Свод знаний по управлению проектами, 1996, 2000, 2004 гг.
- ❑ Области знаний управления проектами:
  1. Управление интеграцией - Project Integration Management
  2. Управление ограничениями - Project Scope Management
  3. Управление временем - Project Time Management
  4. Управление затратами - Project Cost Management
  5. Управление рисками - Project Risk Management
  6. Управление персоналом - Project Personnel Management
  7. Управление коммуникациями - Project Communication Management
  8. Управление закупками - Project Procurement Management
  9. Управление качеством - Project Quality Management



- ❑ IEEE Computer Society Software Engineering Body of Knowledge – Свод знаний по программной инженерии, 18 мая 2004 г.
- ❑ Области знаний программной инженерии:
  1. Software Requirements – требования к ПО
  2. Software Design – проектирование ПО
  3. Software Construction – конструирование ПО
  4. Software Testing – тестирование ПО
  5. Software Maintenance – сопровождение ПО
  6. Software Configuration Management – управление конфигурац.
  7. Software Engineering Management – управление IT проектом
  8. Software Engineering Process – процесс ПИ
  9. Software Engineering Tools and Methods – методы и инструменты
  10. Software Quality – качество ПО

# ACM/IEEE Computing Curricula



- ❑ ACM/IEEE Computing Curricula 2001 – Академический образовательный стандарт в области компьютерных наук 2001
- ❑ Основные разделы компьютерных наук:
  - Computer science – Информатика (2001г);
  - Computer engineering – Компьютерная инженерия;
  - Software engineering – Программная инженерия (2004г.)
  - Information systems – Информационные системы.
- ❑ Сайты:
  - CC2001: <http://www.computer.org/education/cc2001>
  - CC2001. Информатика: <http://se.math.spbu.ru/cc2001>





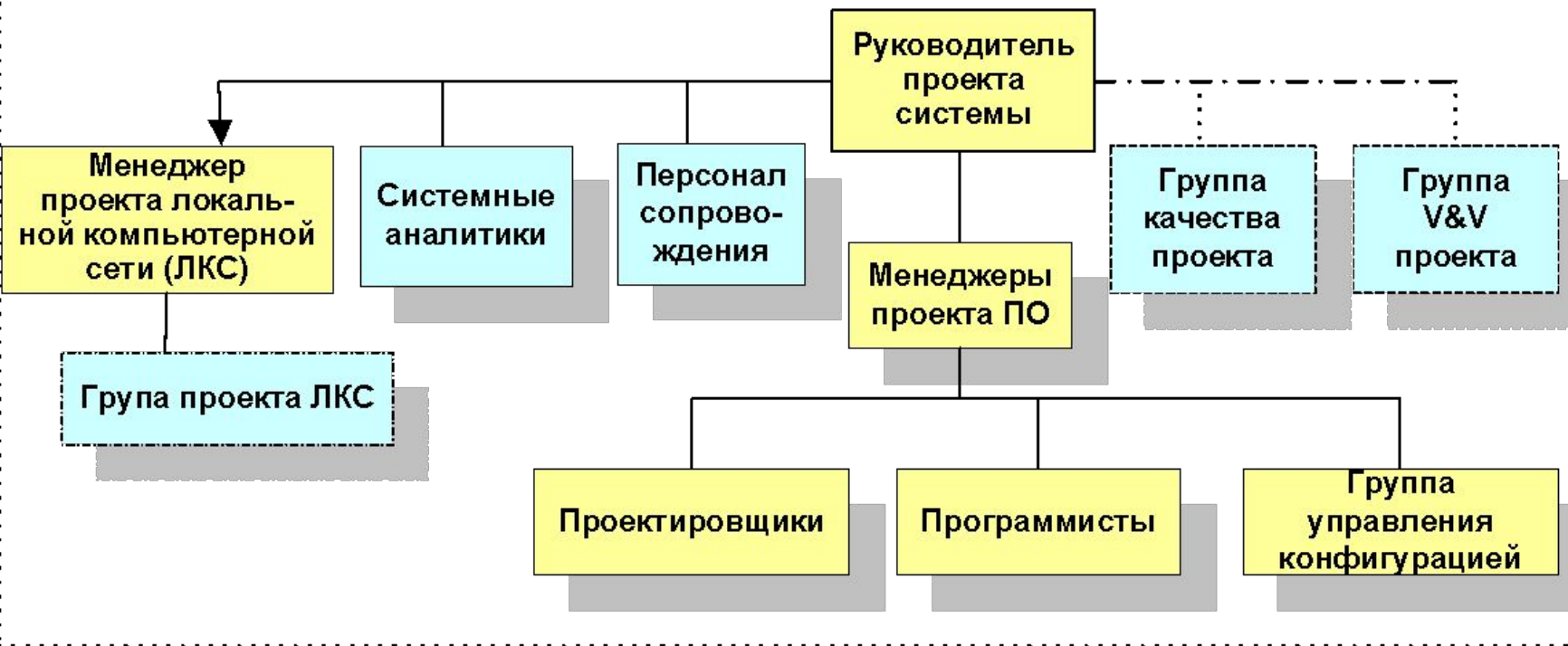
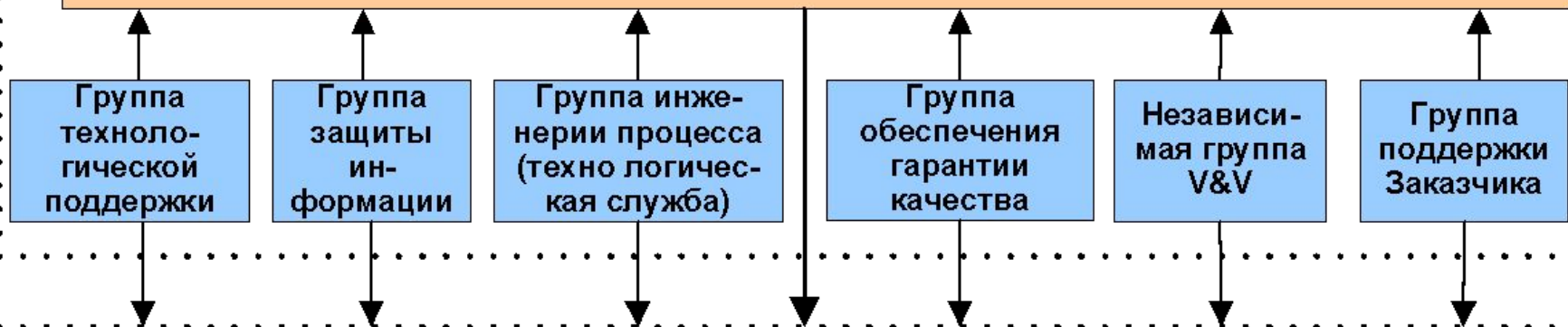
- Области знаний Software engineering:
  - Computing Essentials - Основы применения ЭВМ
  - Mathematical & Engineering Fundamentals - Математические и инженерные основы
  - Professional Practice - Профессиональная практика
  - Software Modeling & Analysis - Моделирование и анализ ПО
  - Software Design - Проектирование ПО
  - Software V & V –Верификация и валидация ПО
  - Software Evolution - Эволюция ПО
  - Software Process - Процесс ПО
  - Software Quality - Качество ПО
  - Software Management -Управление проектом

---

**Артефакт** — это любой созданный искусственно элемент программной системы. К элементам программной системы, а, следовательно, и к **артефактам**, могут относиться исполняемые файлы, исходные тексты, веб страницы, справочные файлы, сопроводительные документы, файлы с данными, модели и многое другое, являющееся физическим носителем информации.

---

# Организационная структура фабрики



# Примеры фабрик программ

1. Система АПРОП (ИК), которая работала в среде ОС ЕС и объединяла разноязыковые модули через интерфейсных посредников методом сборки;
2. Система Sun Microsystems (IBM) со сборкой разноязыковых программ и новыми направлениями производства сложных ПП по модели SOA, Web-сервисы, Ruby, Script и др.;
3. ОМА-архитектура или система CORBA (OMG) обеспечивает взаимодействие клиента и сервера через модули-посредники Stub (для клиента), skeleton, Dll (для сервера), которые передают внешние данные брокеру для выполнения их клиентом или сервером;
4. Фабрика «ручной» сборки разноязыковых программ Инга Бейя с использованием интерфейсных посредников, конфигурационных файлов в средах (VC++, VBasic, Matlab, Java, Visual Works Smalltalk и др.);
5. Фабрики программ для бизнес-программ в UML и MDA - Дж.Гринфильда;
6. Коллективная сетевая среда MS.VSTS для производства программ и ПП разного назначения по контрактам специалистов разных стран мира;
7. Фабрика программ Г.Ленца по схеме производства программ в .Net;
8. Инфраструктура системы Grid – тестирование, сборка и сертификация научных программ, ПП для вычисления в международной сети Европроекта.