ПОНЯТИЕ ПРОГРАММНОЙ ИНЖЕНЕРИИ

Тема 1.1

Преподаватель: Аюшеева Наталья Николаевна

Первый кризис программирования

- Программная инженерия (промышленное программирование) обычно ассоциируется с разработкой больших и сложных программ коллективами разработчиков
- Проблемы становления и развития отрасли деятельности – высокая стоимость программного обеспечения, сложность его создания, необходимость управления и прогнозирования процессов разработки
- Конец 60-х – начало 70-х годов прошлого века – первый кризис программирования, который проявился в том. что стоимость программного обеспечения стала приближаться к стоимости аппаратуры («железа»)
- Цель программной инженерии – сокращение стоимости программ

История

- Термин – software engineering (программная инженерия) - впервые был озвучен в октябре 1968 года на конференции подкомитета НАТО по науке и технике (г. Гармиш, Германия). Рассматривались проблемы проектирования, разработки, распространения и поддержки программ. Там впервые и прозвучал термин «программная инженерия» как некоторая дисциплина, которую надо создавать и которой надо руководствоваться в решении перечисленных проблем
- Позже в Лондоне состоялась встреча 22-х руководителей проектов по разработке ПО. Применяющиеся принципы и методы разработки ПО требовали постоянного усовершенствования. На встрече была предложена концепция жизненного цикла ПО (SLC – Software Lifetime Cycle) как последовательности шагов-стадий, которые необходимо выполнить в процессе создания и эксплуатации ПО.
- В 1970 г. У.У. Ройс (W.W. Royce) произвел идентификацию нескольких стадий в типичном цикле и было высказано предположение, что контроль выполнения стадий приведет к повышению качества ПО и сокращению стоимости разработки.

Предпосылки

- ***Повторное использование кода (модульное программирование)***
- ***Рост сложности программ (структурное программирование)***
- ***Модификация программ (ООП)***

Модульное программирование

- Проблема. На первых этапах становления программной инженерии было отмечено, что высокая стоимость программ связана с разработкой одинаковых (или похожих) фрагментов кода в различных программах. Вызвано это было тем, что в различных программах как части этих программ решались одинаковые (или похожие) задачи: решение нелинейных уравнений, расчет заработной платы, … Использование при создании новых программ ранее написанных фрагментов сулило существенное снижение сроков и стоимости разработки.
- Главный принцип модульного программирования состоял в выделении таких фрагментов и оформлении их в виде модулей. Каждый модуль снабжался описанием, в котором устанавливались правила его использования – интерфейс модуля. Интерфейс задавал связи модуля с основной программой – связи по данным и связи по управлению. При этом возможность повторного использования модулей определялась количеством и сложностью этих связей, или насколько эти связи удалось согласовывать с организацией данных и управления основной программы. Наиболее простыми в этом отношении оказались модули решения математических задач: решения уравнений, систем уравнений, задач оптимизации. К настоящему времени накоплены и успешно используются большие библиотеки таких модулей. Для многих других типов модулей возможность их повторного использования оказалась проблематичной в виду сложности их связей с основной программой. Например, модуль расчета зарплаты, написанный для одной фирмы, может не подойти для другой, т.к. зарплата в этих фирмах рассчитывается не во всем одинаково. Повторное использование модулей со сложными интерфейсами является достаточно актуальной и по сей день. Для ее решения разрабатываются специальные формы (структуры) представления модулей и организации их интерфейсов.

Структурное программирование

- Проблема. Следующий этап возрастания стоимости ПО был связан с переходом от разработки относительно простых программ к разработке сложных программных комплексов. Сложность таких комплексов оценивалась следующими показателями:
  - Большой объем кода (миллионы строк)
  - Большое количество связей между элементами кода
  - Большое количество разработчиков (сотни человек)
  - Большое количество пользователей (сотни и тысячи)
  - Длительное время использования
- Для таких сложных программ оказалось, что основная часть их стоимости приходится не на создание программ, а на их внедрение и эксплуатацию. По аналогии с промышленной технологией стали говорить о жизненном цикле программного продукта, как о последовательности определенных этапов: этапа проектирования, разработки, тестирования, внедрения и сопровождения.

Структурное программирование

- Этап сопровождения программного комплекса включал действия по исправлению ошибок в работе программы и внесению изменений в соответствии с изменившимися требованиями пользователей.
- Основная причина высокой стоимости этапа сопровождения – программы были плохо спроектированы – документация была не понятна и не соответствовала программному коду, а сам программный код был очень сложен и запутан.
- Таким образом, нужна была технология, которая обеспечит «правильное» проектирование и кодирование.
- Основные принципы технологии структурного проектирования и кодирования:
  - Нисходящее функциональное проектирование, при котором в системе выделяются основные функциональные подсистемы, которые потом разбиваются на подсистемы и т.д. (принцип «разделяй и властвуй»)
  - Применение специальных языков проектирования и средств автоматизации использования этих языков
  - Дисциплина проектирования и разработки: планирование и документирование проекта, поддержка соответствие кода проектной документации
  - Структурное кодирование без goto

Объектно-ориентированное программирование

- Проблема. Следующая проблема роста стоимости программ была вызвана тем, что изменение требований к программе стали возникать не только на стадии сопровождения, но и на стадии проектирования – проблема заказчика, который не знает, что он хочет. Создание программного продукта превратилось в его перманентное перепроектирование. Возник вопрос, как проектировать и писать программы, чтобы обеспечить возможность внесений изменений в программу, не меняя ранее написанного кода.
- Решением этой проблемы стало использование подхода или метода, который стали называть объектно-ориентированным проектированием и программированием. Суть подхода состоит в том, что вводится понятие класса как развитие понятия модуля с определенными свойствами и поведением, характеризующими обязанностями класса. Каждый класс может порождать объекты – экземпляры данного класса. При этом работают основные принципы (парадигмы) ООП:
  - Инкапсуляция – объединение в классе данных (свойств) и методов (процедур обработки).
  - Наследование – возможность вывода нового класса из старого с частичным изменением свойств и методов
  - Полиморфизм – определение свойств и методов объекта по контексту

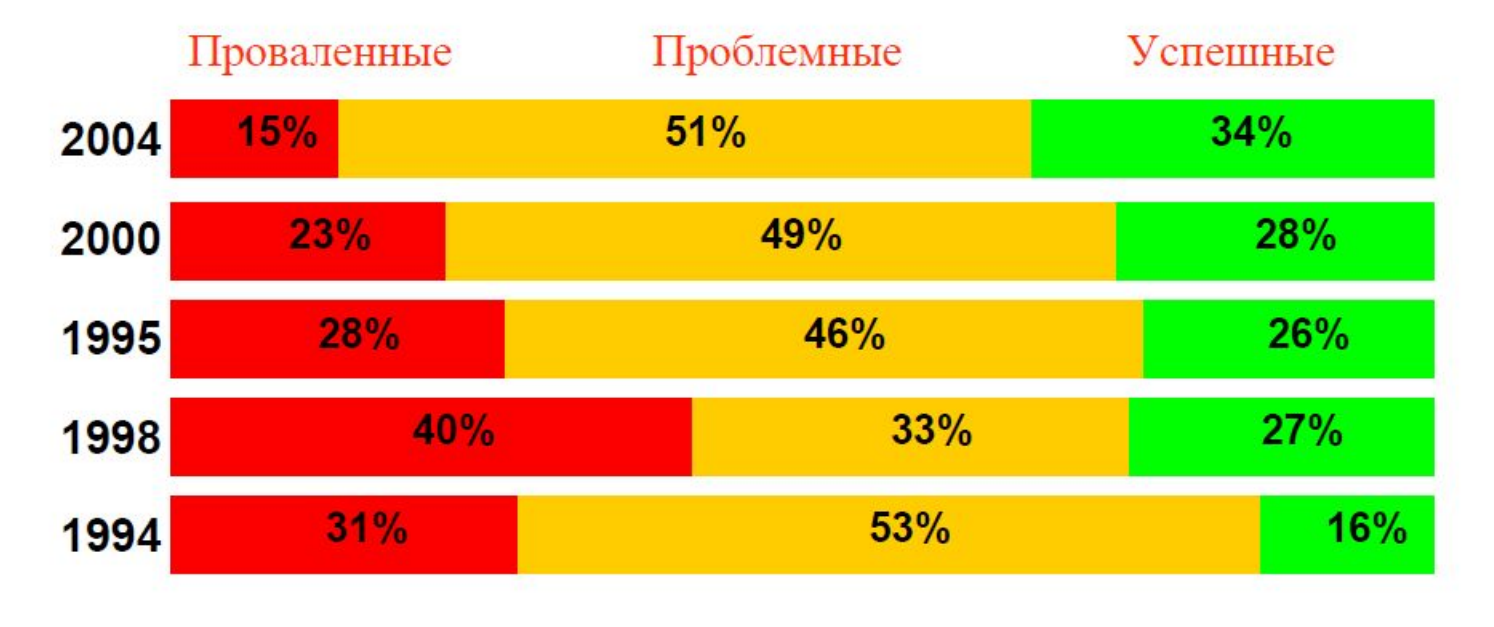
Некоторые выводы

- Программная инженерия (или технология промышленного программирования) как некоторое направление возникло и формировалось под давлением роста стоимости создаваемого программного обеспечения.
- Главная цель этой области знаний – сокращение стоимости и сроков разработки программ.
- Программная инженерия прошла несколько этапов развития, в процессе которых были сформулированы фундаментальные принципы и методы разработки программных продуктов.
- Основной принцип программной инженерии состоит в том, что программы создаются в результате выполнения нескольких взаимосвязанных этапов (анализ требований, проектирование, разработка, внедрение, сопровождение), составляющих жизненный цикл программного продукта.
- Фундаментальными методами проектирования и разработки являются модульное, структурное и объектно-ориентированное проектирование и программирование.

Второй кризис программирования

- Рубеж 80–90-х годов отмечается как начало информационно-технологической революции, вызванной взрывным ростом использования информационных средств: персональный компьютер, локальные и глобальные вычислительные сети, мобильная связь, электронная почту, Internet и т.д.
- США тратит ежегодно более $200 млрд. на более чем 170 тыс. проектов разработки ПО в сфере IT; 31,1% из них закрываются, так и не завершившись; 52,7% проектов завершаются с превышением первоначальных оценок бюджета/сроков и ограниченной функциональностью; потери от недополученного эффекта внедрения ПО измеряются триллионами.

Статистика по 30 000 проектам по разработке ПО в американских компаниях



Успешные проекты – вовремя и в рамках бюджета был выполнен весь намеченный фронт работ

Проблемные проекты – нарушение сроков, перерасход бюджета и/или сделали не все, что требовалось

Проваленные проекты – не были доведены до конца из-за перерасхода средств, бюджета, качества.

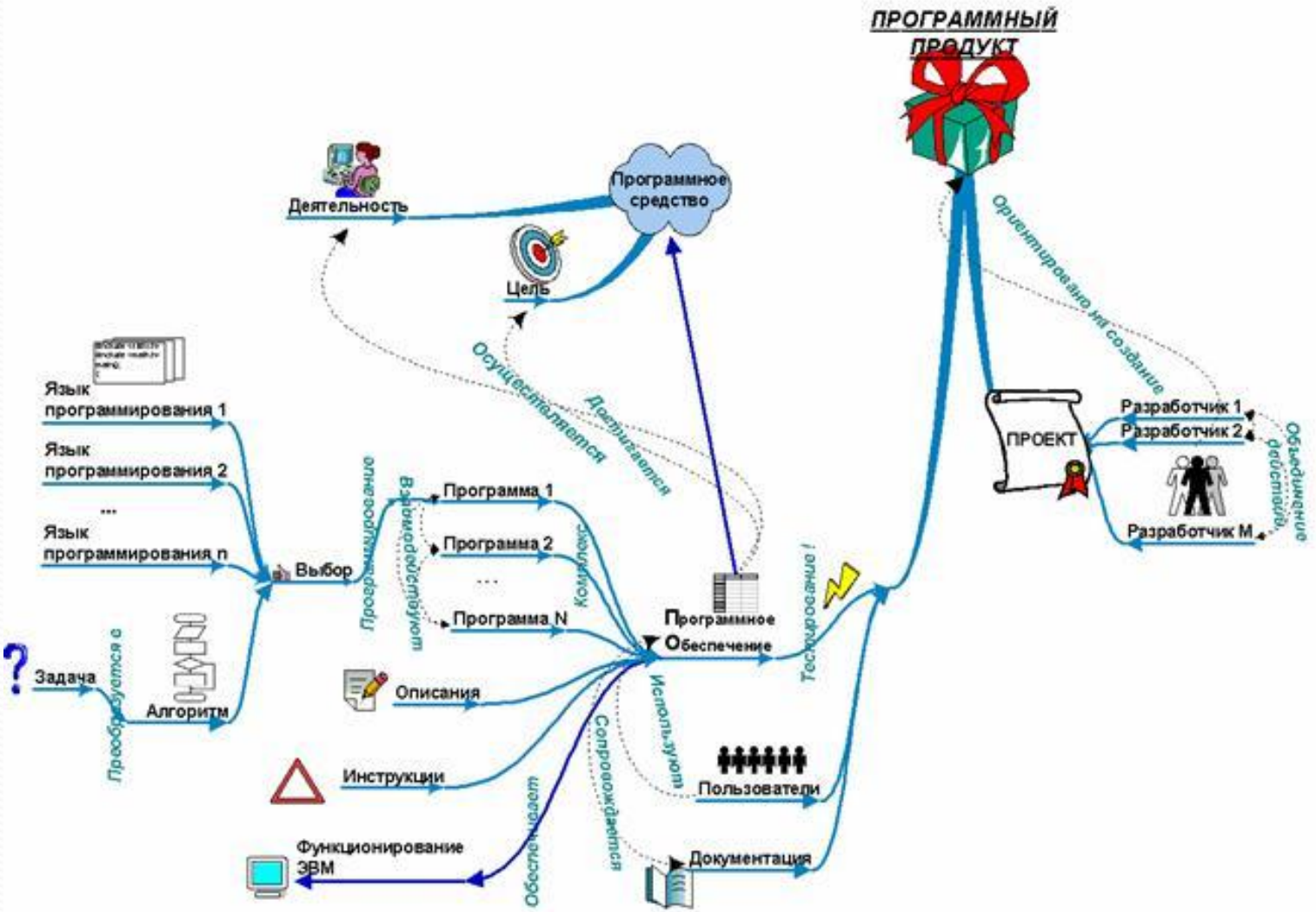
Определения понятия «программная инженерия»

- 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].

Что такое программная инженерия ?

- Что такое программное обеспечение (software)?
- Что такое программная инженерия?
- В чем отличие программной инженерии от информатики (computer science)?
- В чем отличие программной инженерии от других инженерий?
- Что такое методы программной инженерии?
- Что такое CASE (Computer-Aided Software Engineering)?
- Какими свойствами обладает хорошая программа?

Что такое программное обеспечение ?



Что такое программная инженерия?

- Программная инженерия — это инженерная дисциплина, которая связана со всеми аспектами производства ПО от начальных стадий создания спецификации до поддержки системы после сдачи в эксплуатацию.
- Инженерная дисциплина. Инженеры – это те специалисты, которые выполняют практическую работу и добиваются практических результатов. Инженеры работают в условиях ограниченных ресурсов: временных, финансовых и организационных (оборудование, техника, люди). Иными словами, продукт должен быть создан в установленные сроки, в рамках выделенных средств, оборудования и людей.
- Все аспекты производства ПО. Программная инженерия занимается не только техническими вопросами производства ПО (специфицирование требований, проектирование, кодирование,...), но и управлением программными проектами, включая вопросы планирования, финансирования, управления коллективом и т.д. Кроме того, задачей программной инженерии является разработка средств, методов и теорий для поддержки процесса производства ПО.

В чем отличия от информатики?

- Информатика (computer science) занимается теорией и методами вычислительных и программных систем, в то время как программная инженерия занимается практическими проблемами создания ПО. Информатика составляет теоретические основы программной инженерии и инженер по программному обеспечению должен знать информатику.
- Информатика – это не единственный теоретический фундамент программной инженерии, т.к. круг проблем, стоящих перед программным инженером значительно шире просто написания программ. Это еще управление финансами, организация работ в коллективе, взаимодействие с заказчиком и т.д.

В чем отличие от других инженерий?

- Отличие программной инженерии от других инженерий интересно прежде всего с точки зрения двух вопросов:
  - Почему доля провальных проектов в программной инженерии так велика по сравнению с другими инженериями?
  - Можно ли в программной инженерии применять опыт других инженерий?

Методы программной инженерии

- Метод программной инженерии — это структурный подход к созданию ПО, который способствует производству высококачественного продукта эффективным в экономическом аспекте способом. В этом определении есть две основные составляющие: (а) создание высококачественного продукта и (б) экономически эффективным способом.
- Начиная с 70-х годов создано достаточно много методов разработки ПО. Наиболее известны:
  - Метод структурного анализа и проектирования Том ДеМарко (1978)
  - Метод сущность-связь проектирования информационных систем Чен (1976)
  - Метод объектно-ориентированного анализа Буч (1994), Рамбо (1991)
- Метод программной индустрии основан на идее создания моделей ПО с поэтапным преобразованием этих моделей в программу – окончательную модель решаемой задачи. Так, на этапе спецификаций создается модель – описание требований, которая далее преобразуется в модель проекта ПО, проект – в программный код. При этом важно, чтобы модели метода представлялись графически с помощью некоторого языка представления моделей.
- Методы должны включать в себя следующие компоненты:
  - Описание моделей системы и нотация, используемая для описания этих моделей (например, объектные модели, конечно-автоматные модели и т.д.)
  - Правила и ограничения, которые надо выполнять при разработке моделей (например, каждый объект должен иметь одинаковое имя)
  - Рекомендации — эвристики, характеризующие хорошие приемы проектирования в данном методе (скажем, рекомендация о том, что ни у одного объекта не должно быть больше семи подобъектов)
  - Руководство по применению метода — описание последовательности работ (действий), которые надо выполнить для построения моделей (все атрибуты должны быть задокументированы до определения операций, связанных с этим объектом)

Что такое CASE?

- CASE - Computer Aided System Engineering - различного рода инструментальные программы, используемые для поддержки процесса создания программ
- CASE средства могут быть классифицированы по нескольким признакам:
  - По уровню применения:
    - Upper CASE -средства анализа требований
    - Middle CASE - средства проектирования
    - Low CASE - средства разработки приложений
  - Специализированные
    - Средства проектирования баз данных
    - Средства реинжиниринга (восстановления) модели (формирование ERD на основе анализа схем БД или формирования диаграмм на основе анализа программных кодов)
  - Вспомогательные
    - Планирования и управления проектом
    - Конфигурационного управления
    - Тестирования
  - Интегрированные CASE охватывают все этапы и процессы создания ПО от анализа требований до тестирования и выпуска документации. Интегрированные CASE выступают, как правило, в виде набора согласованных по интерфейсе средств, предназначенных для поддержки отдельных этапов процесса.
- При выборе CASE средств следует руководствоваться основным принципом: сначала метод создания ПО, а потом – CASE средства, применимые для этого метода.

Свойства хорошей программы

- Выполнение функциональных требований
- Соответствие нефункциональным требованиям:
  - Сопровождаемость (maintainability)
  - Надежность (dependability)
  - Эффективность (efficiency)
  - Удобство использования (usability)

Профессиональные и этические требования

- Специалисты по программному обеспечению работают в определенном правовом и социальном окружении, находятся под действием международных, национальных и местных законодательств.
- Но вместе с тем, программисты не могут руководствоваться только моральными нормами или юридическими ограничениями, т.к. они обычно бывают связаны более тонкими профессиональными обязательствами:
  - Конфиденциальность – программные специалисты должны уважать конфиденциальность в отношении своих работодателей или заказчиков независимо от того, подписывалось ли ими соответствующее соглашение.
  - Компетентность – программный специалист не должен завышать свой истинный уровень компетентности и не должен сознательно браться за работу, которая этому уровню не соответствует.
  - Защита интеллектуальной собственности – специалист должен соблюдать законодательство и принципы защиты интеллектуальной собственности при использовании чужой интеллектуальной собственности. Кроме того, он должен защищать интеллектуальную собственность работодателя и клиента.
  - Злоупотребление компьютером – программный специалист не должны злоупотреблять компьютерными ресурсами работодателя или заказчика

Кодекс этики IEEE-CS/ACM

- Кодекс этики и профессиональной практики программной инженерии
- Разработчики: ACM (Association for Computing Machinery – Ассоциация по вычислительной технике), IEEE (Institute of Electrical and Electronic Engineers – Институт инженеров по электротехнике и электронике) и BCS (British Computer Society – Британское компьютерное общество)
- 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.
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.
7. 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.