**Предмет и основные понятия программной инженерии**

Инженер – специалист с высшим техническим образованием

Инженерия – область технической деятельности, направленная на практическое применение знаний.

Профессиональная деятельность инженеров связана с проектированием, производством или эксплуатацией сложных технических объектов.

Базовые характеристики, которые используются для описания основ инженерии:

1. Инженерная деятельность неразрывно связана с необходимостью принятия решений и выборов подходов
2. Инженеры несут ответственность за выполнение широкого спектра задач
3. Инженеры используют измеримые количественные характеристики
4. Инженеры в своей деятельности используют инструментальные средства
5. Инженеры повторно используют результаты проектирования
6. Инженеры используют дисциплинированные процессы
7. Инженеры объединяются в профессиональные сообщества

Программная инженерия –

* это установление и использование обоснованных инженерных принципов для экономного получения ПО, которое надежно и работает на реальных машинах.
* система методов, способов и дисциплин планирования, разработки эксплуатации и сопровождающего ПО, предназначенного для промышленного производство ПО
* это дисциплина, целью которой является создание качественного ПО, которое завершается вовремя, не превышает выделенных бюджетных средств и удовлетворяет выдвигаемым требованиям

Программа – это объект разработки, который доступен пониманию ЭВМ, для которой написан

Готовая программа – программный продукт (ПП), реализующий определенные функции предметной области

Программный продукт (ПП) – это комплекс взаимосвязанных программ для решения определенной проблемы массового спроса, подготовленный к реализации как любой вид промышленной продукции

Производство ПП базируется на технологических процессах изготовления определенных видов продуктов с применением теории проектирования и использования инструментальных средств

Метод разработки – это способ или планомерный подход к достижению той цели, которая ставится перед созданием объекта разработки. Наиболее распространенным методом является метод модульного программирования, обеспечивающий декомпозицию задачи на отдельные функции, вплоть до элементарной, каждая из которых представляется модулем или программой.

Технологический процесс – это взаимосвязанная последовательность операций, выполняемых при разработке объекта. Процесс предназначен для перевода объекта из одного состояния в другое вплоть до получения конечного продукта

Инструмент (CASE-средство) – это программное или методическое средство для получения объекта в законченном виде. Это трансляторы, отладчики, генераторы, сборщики, тестеры и т.п.

Программная инженерия – это практическое применение научных знаний при разработке и создании компьютерных программ и связанной с ними документации, необходимой для их разработки, использования и поддержки

Основу программной инженерии составляют классические науки: теория алгоритмов, теория множеств, теория доказательств, математическая логика, теория программирования, теория абстрактных данных, теория управления и др.

Базовые понятия и объекты программной инженерии составляют: структуры данных, функции и композиции, формализмы для описания систем теория преобразования данных для организации вычислений

Возникновение программной инженерии как дисциплины разработки ПО определено многими факторами, среди которых самыми важными являются следующие:

1. Накопление в области создания ПО значительного объема разнородных знаний
2. Появление новых методов проектирования и разработки ПО
3. Совершенствование

Набор инженерных приемов, средств и стандартов, ориентированных на изготовление целевых ПП сосредоточен в проект ***Software Engineering Body of Knowledge*** (SWEBOK)

**Руководство к Своду Знаний по Программной Инженерии**

***SWEBOK*** включает базовое определение и описание областей знаний и является необходимым для понимания вопросов разработки ПО

Одной из важнейших целей ***SWEBOK*** является именно определение тех аспектов деятельности, которые составляют суть профессии инженера-программиста

*SWEBOK* включает 10 областей знаний (*knowledge areas, КА*), которые соответствуют процессам проектирования ПО и методам их поддержки, а именно:

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 конфигурацией
7. Software engineering management – управление в программной инженерии
8. **Software engineering process** – процессы программной инженерии
9. **Software engineering tools and methods** – инструменты и методы программной инженерии
10. **Software quality** – качество ПО

**Современный инженер-программист – это специалист:**

* Выполняющий практические работы по созданию программного обеспечения
* Способный обеспечить высокое качество разработки за счет применения современных методов проектирования, использования готовых компонентов и методов их генерации
* Знакомый со стандартами, регламентирующими процесс разработки ПО
* Уверенно оперирующий понятием «жизненные цикл ПО»
* Хорошо владеющий навыками использования инструментальных систем программирования
* Знающий методы управления проектами, может произвести оценку качества, производительности, стоимости и т.п.
* Умеет эффективно работать в команде

Программная инженерия – это область деятельности, где кроме **программистов**, занимающихся непосредственно разработкой ПП, в программной инженерии задействованы:

* **Менеджеры**, которые планируют и руководят проектом, отслеживают сроки его исполнения и затраты
* **Инженеры службы ведения** библиотеки и репозиториев компонентов
* **Технологии,** которые проверяют правильность выполнения процесса разработки ПП путем тестирования, и на основе собранных данных проводят измерения характеристик качества
* **Валидаторы**, которые проверяют ПП на соответствие заданным требованиям
* **Верификаторы**, которые проверяют правильность реализации алгоритмов и программ в проекте путем их сопоставления с эталонными данными, алгоритмам или программами

Программная инженерия является развивающейся дисциплиной

Программная инженерия не касается вопросов конкретизации применения тех или иных языков программирования

**Программирование** – раздел информатики, изучающий методы и приемы составления программ для компьютеров

**Программирование** (в широком смысле) – все технические операции, необходимые для создания программы, включая анализ требований и все стадии разработки и реализации

**Программирование** (в узком смысле) – процесс кодирования и отладки программы в рамках реального проекта

**Целью программирования** является описание процессов обработки данных (в дальнейшем – просто процессов)

**Программа** – это набор операторов, который может быть представлен как единое целое в некоторой вычислительной системе и который используется для управления поведением этой системы

**Программирование** – процесс отображения определенного множества целей на множество машинных команд и данных, интерпретация которых на компьютере или вычислительном комплексе обеспечивает достижение поставленной цели

**Профессиональное программирование (производство программ)** – деятельность, направленная на получение доходов при помощи программирования

**Профессиональный программист** – человек, который занимается профессиональным программированием

**Технология = набор правил + методик + инструментов + процессы планирования, оценки качества + др.**, позволяющих наладить производственный процесс выпуска продукта, сокращая его стоимость и повышая качество

**Технологией программирования = программной инженерией**

**Программная инженерия**, определяемой как систематический подход к разработке, эксплуатации, сопровождению и изъятию из обращения программных средств

**Стандарты и проблемы жизненного цикла ПО**

**Современный инженер-программист – это специалист**

**Стандарт** является нормативно-техническим документом, утверждаемым компетентным органом, устанавливающим комплекс норм, правил по отношению к предмету стандартизации, или типовым образцом, эталоном, моделью, которые принимаются как исходные для сопоставления с ними других предметов

Стандарт может быть разработан как на материально-технические предметы (продукции, эталоны, образцы веществ), так и на нормы, правила, требования организационно-методического и общественного характера

Пример, вузы работают в соответствии с государственными образовательными стандартами, представленными в виде паспортов специальностей

**Стандартизация** – это деятельность по разработке (введению), утверждению, изменению (актуализации), отмене, опубликованию и применению документов по стандартизации и иная деятельность, направленная на достижение упорядоченности в отношении объектов стандартизации

В настоящее время принято выделять следующие **основные типы стандартов**

**Корпоративные стандарты** разрабатывают крупными фирмами (корпорации) с целью повышения качества своей продукции. Такие стандарты разрабатываются на основе собственного опыта и с учетом требований мировых стандартов.

Корпоративные стандарты не сертифицируются, но являются обязательными для применения внутри корпорации

В условиях рыночной конкуренции могут иметь закрытый характер

В сфере информационных технологий известны стандарты, разработанные Microsoft, Intel, IBM

**Отраслевые стандарты** действуют в пределах организаций некоторой отрасли (министерства). Например, СНИП – строительные нормы и правила. Разрабатываются с учетом требований мирового опыта и специфики отрасли. Являются, как правило, обязательными для отрасли. Подлежат сертификации. Язык SQL – стандарт языка обращения к базам данных.

**Государственные стандарты (ГОСТы)** принимаются государственными органами, имеют силу закона. Разрабатываются с учетом мирового опыта или на основе отраслевых стандартов. Могут иметь как рекомендательный, так и обязательный характер (стандарты безопасности). Для сертификации создаются государственные или лицензированные органы сертификации.

**Международные стандарты.** Разрабатываются, как правило, специальными международными организациями на основе мирового опыта и лучших корпоративных стандартов. Имеют сугубо рекомендательный характер. Право сертификации получают организации (государственные и частные), прошедшие лицензирование в международных организациях.

В области информационных технологий существуют определенные проблемы создания стандартов:

* Острая конкуренция ИТ-компаний
* Недостаток культуры совместной разработки стандартов силами профессиональных сообществ
* Отсутствие национального рынка стандартов и конкуренции между стандартами
* Высокая степень неопределенности проектов разработки и внедрения ИТ
* Скорость развития – стандарты не успевают «созреть», а уже требуются новые
* Сложность согласования стандартов, относящихся к одному и тому же предмету, но утвержденных разными стандартизирующими организациями
* Отсутствие общего терминологического словаря
* Нет развитой системы независимого тестирования и сертификации программных продуктов (имеющиеся лаборатории немногочисленны, находятся при ведомственных институтах и т.д.)
* Очень существенным является «языковой барьер»

*Недостаток стандартов в области Программной инженерии – стандартом становится не самое сильное, а самое массовое коммерческого решение*

***Кто разрабатывает стандарты программной инженерии?***

Основными разработчиками международных стандартов являются следующие организации

**ISO** – International Organization for Standardization – **Международная организация по стандартизации.** Наиболее представительная и влиятельная организация, разрабатывающая стандарты почти во всех областях деятельности, в том числе и в IT

**ACM –** Association for Computing Machinery – **Ассоциация по вычислительной технике.** Всемирная научная и образовательная организация в области вычислительной технике. Известна также и разработкой образовательных стандартов

**SEI –** Software Engineering Institute – **Институт Программной Инженерии.** Исследования в области программной инженерии с упором на разработку методов оценки и повышении качества ПО. Стандарты по качеству ПО и зрелости организаций, разрабатывающих ПО

**IEEE – Институт инженеров по электронике.** Поддержка научных и практических разработок в области электроники и вычислительной техники. Самая активная организация в разработке стандартов в этой области

В Российской Федерации в области ИТ действует свыше 700 межгосударственных (ГОСТ) и национальных (ГОСТ Р) стандартов (в мире – свыше 2000)

**Спецификация (стандарт, определение) языка программирования** – это предмет документации, который определяет язык программирования, чтобы пользователи и разработчики языка могли согласовывать, что означают программы на данном языке

Спецификации обычно являются подробными и формальными и в основном используются разработчиками языка, в то время как пользователи обращаются к ним в случае двусмысленности

Сопутствующая документация включает справочник по языку программирования, который специально предназначен для пользователей, и логическое обоснование языка программирования, которое объясняет, почему спецификация написана именно так; последние обычно более неформальны, чем спецификации

Состав документации и содержание программных документов

**Технический писатель** или разработчик пользовательский (и иной) документации как части программного продукта

Функциями технического писателя являются:

* Разработка плана документирования, который включает состав, сроки подготовки и порядок тестирования документов
* Выбор и разработка стандартов и шаблонов подготовки документов
* Выбор средств автоматизации документирования
* Разработка документации
* Организация тестирования документации
* Участие в тестировании продукта. Технический писатель все время работает с продуктом (его готовыми версиями) и выступая от имени пользователя видит все недочеты и несоответствия

В стандарте **ИСО/МЭК 6592:2000** определяется, что существует четыре основные цели документирования:

1. Описывать и записывать информацию о системе на протяжении ее жизненного цикла
2. Способствовать простоте использования и удобству сопровождения ИС
3. Содействовать контролю нал процессом жизненного цикла
4. Сообщать информацию о системе тем, кто в ней нуждается

В стандарте ГОСТ 19.101-77 «Единая система программной документации. Виды программ и программных документов» все документы делятся на две группы: **программные и эксплуатационные**

**Программные документы:**

* Спецификация
* Ведомость держателей подлинников
* ###

**Эксплуатационные документы:**

* Ведомость эксплуатационных документов
* Формуляр
* Описание применения
* Руководство системного программиста
* Руководство программиста
* Руководство оператора
* Описание языка
* Руководство по техническому обслуживанию

ГОСТ 19.201-78 **«Техническое задание. Требования к содержанию и оформлению»** описан порядок построения и оформления технического задания

В частности, устанавливается, что техническое задание должно содержать следующие разделы:

* Введение
* Основания для разработки; назначение разработки
* Требования к программе или программному изделию
* Требования к программной документации
* Технико-экономические показатели
* Стадии и этапы разработки; порядок контроля и приемки
* В техническое задание допускается включать приложения

**«Руководство пользователя»** относится к пакету эксплуатационной документации. Основная цель руководства пользователя заключается в обеспечении пользователя необходимой информацией для самостоятельной работы с программной или автоматизированной системой

ГОСТ 19.505-79 «Руководство оператора. Требования к содержанию и оформлению»

В частности, устанавливается, что руководство пользователя должно содержать следующие разделы:

* Назначение системы
* Условия применения системы
* Подготовка системы к работе
* Описание операции
* Аварийные ситуации

*Технология* – совокупность производственных процессов в определенной отрасли производства

*Технология программирования* – совокупность методов и средств, используемых в процессе разработки программного обеспечения

Она представляет собой набор технологических инструкций, включающих в себя:

* Указание последовательности выполнения технологических операций
* Перечисление условий, при которых выполняются операции
* Описания самих операций с определением исходных данных, результатов, инструкций, нормативов, стандартов, критериев и методов оценки

Непрерывное увеличение объемов и сложности программных комплексов, а также рост требований к их качеству привели к тому, что простейшие технологии программирования небольших программ в настоящее время развились в ***сложные технологии*** проектирования, разработки, и сопровождения интегрированных комплексов программ.

Технологии программирования чаще всего не связаны с конкретным языком

В программной инженерии изучаются различные методы и инструментальные средства разработки (программного продукта) ПП с точки зрения достижения определенных целей – эти методы и средства могут использоваться в разных технологических процессах (и в разных технологиях программирования)

Рассматривая ту или иную технологию программирования, мы будет

* Рассматривать все процессы разработки ПП, начиная с момента возникновения замысла ее создания до написания необходимой документации (программной, системной, для пользователя)
* Рассматривать не только вопросы построения программных конструкций и выбора структур данных, но и вопросы описания функций и принимаемых решений с точки зрения их человеческого (неформального) восприятия
* Использовать ряд базовых принципов для достижения поставленной цели – выпуска надежной ПП

Такой взгляд на технологию программирования существенно влиять на организацию технологических процессов, на выбор в них методов и инструментальных средств

**Парадигма программирования** – это совокупность идей и понятий, определяющих стиль написания компьютерных программ (подход к программированию)

**Парадигма программирования** – модель или подход к решению проблемы

Слово **«парадигма»** используется в программировании для определения семейства обозначений (нотаций), разделяющих общий способ (методику) реализаций программ

Все современные языки программирования предлагают эквивалентные синтаксические конструкции, но философия каждого языка отличается

Это именно тот случай, когда использование методологии программирования имеет реальное значение

Если попытаться охарактеризовать современный уровень развития компьютерных и информационных технологий, то первое, на что следует обратить внимание, - это *возрастающая сложность* не только отдельных *физических и программных компонентов*, но и лежащих в основе этих *технологий концепций и идей*

Чтобы разобраться в существующих технологиях программирования и определить основные тенденции их развития, целесообразно рассмотреть эти технологии в историческом контексте, выделяя основные этапы развития программирования как науки

Сделаем краткую характеристику развития программирования по десятилетиям:

В 50-е годы мощность компьютеров (первого поколения) была невелика, а программирование для них велось в основном в машинном коде

Программирование было искусством: программисту необходимо было отслеживать не только последовательность выполняемых операций, но и местоположение данных при программировании

В соответствии с принципом фон Неймана программы и данные хранились в одной и той же памяти (концепция хранимой программы) и выполнялись последовательно на одном процессоре

В это время решались главным образом научно-технические задачи (расчет по формулам), задание на программирование содержало, как правило, достаточно точную постановку задачи

Появление ассемблеров, а затем и языков программирования высокого уровня упростило программирование вычислений, снизив уровень детализации операция, и позволило повысить уровень сложности программ

В этот период использовались *интуитивная технология программирования*

Почти сразу приступали к кодированию программы, при этом задание могло несколько раз изменяться (это сильно увеличивало время ее разработки), минимальная документация оформлялась уже после того, как программа начинала работать

Тем не менее, именно в этот период родилась фундаментальная для технологии программирования концепция **модульного программирования**, ориентированная на преодоление трудностей программирования в машинном коде

60-е годы прошлого столетия – период «стихийного» программирования

В этот период отсутствовало понятие структуры программы, типов данных и т.д. Вследствие этого код получался запутанным, противоречивым. Программирование тех лет считалось искусством

Языки программирования первоначально делала упор на синтаксические аспекты и проблемы записи программ, т.е. на форму

В этот период наблюдалось бурное развитие и широкое использование языков программирования высокого уровня

***Грамматика*** – конечное непустое множество правил, посредством которых можно породить все допустимые предложения языка программирования

Все используемые на практике языки программирования содержат неограниченное число допустимых предложений

***Синтаксис*** – правила образования текстов

Для описания синтаксиса используют формальную систему обозначений Бэкуса-Наура, принятую в программной документации

С ее помощью легко определить, какие именно последовательности будут считаться программами

Система обозначения Бэкуса-Наура

1. <слова или группы слов>
2. ::= – определяется как
3. | – или
4. {} – произвольное количество повторений
5. [] – необязательные символы

Появление в этих языках средств реализации механизма подпрограмм стало революционным. Это позволило создать большие библиотеки расчетных и служебных подпрограмм, которые можно было сохранить и затем использовать в программах

Первоначально программа состояла из основной программы, области глобальных данных и набора подпрограмм, выполняющих обработку либо всех данных, либо только ее части

Такая архитектура повышала *вероятность искажения* части глобальных данных какой-либо подпрограммой, поэтому для повышения надежности программ была предложена новая идея: использовать часть данных как локальные внутри подпрограмм

Основной методологией того времени стала ***методология процедурно-ориентированного программирования***, в основе которой лежала процедурная или алгоритмическая организация структуры программного кода.

Для снижения количества ошибок и повышения надежность при разработке больших программ программную систему стали разбивать на более мелки фрагменты

Основой для такого разбиения стала ***процедурная (алгоритмическая) декомпозиция***, при которой отдельные части программы или модули представляли собой совокупность процедур для решения некоторой совокупности задач

В середине 60-х годов в языках программирования появилась специальная синтаксическая конструкция, и закрепилось новое понятие ***процедуры***

Главная особенность ***процедурного программирования*** заключается в том, что программа всегда имеет начало во времени и окончание, для начала действий последующей процедуры необходимо завершение всех действий предшествующей процедуры

Появление транзисторов, а затем и интегральных схем, стоимость компьютеров резко снизилась, а их производительность росли почти экспоненциально. Появление в компьютерах 2-ого поколения прерываний привело к развитию мультипрограммирования и возможности создания больших программных систем.

Широко стала использоваться коллективная разработка, которая поставила ряд серьезных технологических проблем, **разразился «кризис программирования»**.

* Фирмы, взявшиеся за разработки сложных П, срывали сроки их завершения
* Существенно увеличивалась стоимость таких проектов

Объективно это было вызвано несовершенством технологии программирования

Применяемый в это время подход разработки сложных ПП **«снизу-вверх»**, при котором вначале разрабатывались самые *простые подпрограммы*, а затем из них *конструировались* *более* *сложные*, привел к тому, что при согласовании подпрограмм выявлялось большое количество ошибок, на устранение которых требовалось очень много времени.

Процесс тестирования и отладки стал занимать больше времени, чем процесс кодирования

Появление интенсивное использование условных операторов безусловного перехода *goto* стало предметом острых дискуссий среди специалистов по программированию

Дело в том, что бесконтрольное применение в программе оператора безусловного перехода goto способно серьезно осложнить понимание кода. Ситуация казалась настолько драматичной, что в литературе зазвучали призывы исключить оператор goto из языков программирования.

Именно с этго времени принято считать хорошим стилем программирования – **программирование без goto**

Это стало началом серьезных размышлений над методологией и технологий программирования, стал развиваться новый подход к программированию, который был назван **структурным**

В 70-е годы получили широкое распространение информационные системы (ИС) и базы данных (БД), так как к середине 70-х годов стоимость хранения одного бита информации на компьютерных носителях стала меньше, чем на традиционных носителях. Это резко повысило интерес к компьютерным системам хранения данных.

Базовой методологией стала методология **структурного программирования**, в основе которой лежит и процедурная **декомпозиция ПП и организация отдельных модулей в виде совокупности выполняемых процедур**

В отличие от процедурного, структурный подход требовал представления задаче в виде некоторой иерархии подзадач, для достижения общей идеи проектирование системы должно было вестись по принципу «сверху-вниз» с применением метода пошаговой детализации

Появились языки программирования 3-го поколения (Pascal, Modula-2), отличительной особенностью которых явилось наличие развитых средств абстрагирования типов для структурирования данных

**Появление пользовательских типов** позволило уменьшить количество ошибок при работе с глобальными данными

Стала развиваться технология модульного программирования, которая предполагает объединение нескольких подпрограмм, использующих одни и те же данные, в отдельно компилируемые модули, связи между которыми устанавливались через специальный интерфейс, а доступ к реализации регулировался механизмами импорта-экспорта…

Программы делятся на «большие» (создаются для решения сложных задач, связанных с практической деятельностью значительного числа людей, 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, сопровождаются полной и понятной пользователям документацией, а также специальной документацией для администраторов, дополняется проектной документации к ПС, что позволяет развивать ее даже без привлечения первоначальных разработчиков, такую программу невозможно написать с первой попытки, с небольшими усилиями в одиночку, ее разработка ведется коллективом специалистов) и «маленькие» (сравнительно небольшой размер, направленность на решение одной четко поставленной задачи с хорошо известными ограничениями, отсутствие оптимизации по скорости выполнения, практическое отсутствие ущерба от неправильной работы программы, отсутствие необходимости дополнять программу новыми возможностями и разрабатывать ее новые версии, минимальная потребность в документации).

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

Технология разработки программного обеспечения (ТРПО) – совокупность процессов и методов создания программного продукта

Технология разработки ПО – система инженерных принципов для создания экономичного ПО, которое надежно и эффективно работает в реальных компьютерах

Технология программирования – способ организации **процесс** создания программы, совокупность приемов и способов выполнения определенных видов деятельности

**Разработка = анализ + проектирование + программирование + тестирование + анализ**

Процесса разработки ПО определяет то, какие задачи должны решаться в проекте и в каком порядке они должны выполняться

Жизненный цикл ПО – период времени, который начинается с момента принятия решения о необходимости создания программного продукта и заканчивается в момент его полного изъятия из эксплуатации

Самой простой моделью процесса является модель водопада

В данной модели проект начинается с анализа осуществимости

При успешной демонстрации осуществимости проекта, начинается анализ требования и планирование проекта

После завершения анализа требований начинается проектирование, а после него начинается составление программы

После успешного завершения программирования, созданный код объединяется и выполняется тестирование

После успешного завершения тестирования, система устанавливается у заказчика

После этого выполняется постоянное использование созданного ПО и его поддержка

Анализ требований: наименование задачи, словесное описание: ограничения на значения, точность вычислений, назначение, принципиальная возможность решения поставленной задачи, структура входных и выходных данных; внешние спецификации функции, внешние спецификации интерфейса, внешние данные тестирования, пример работающей программы

Проектирование: уточнение наименования программы, выбор методы решения задачи, уточненные глобальные данные и пользовательские типы, декомпозиция, алгоритмизация

Кодирование

Инструмент поддерживает определенную деятельность

Программной средой – набор связанных между собой инструментов